

CREATING NEW CONCEPT-BASED REPRESENTATIONS FOR SUPERIOR
TEXT ANALYSIS AND RETRIEVAL

by

Walid Shalaby

A dissertation submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in
Computing and Information Systems

Charlotte

2018

Approved by:

Dr. Wlodek Zadrozny

Dr. Srinivas Akella

Dr. Jing Yang

Dr. Zbigniew Ras

Dr. M. Yasin Raja

ABSTRACT

WALID SHALABY. Creating New Concept-based Representations for Superior Text Analysis and Retrieval. (Under the direction of DR. WLODEK ZADROZNY)

Text analytics represent a set of scalable techniques that mine unstructured and semi-structured textual resources in order to extract useful knowledge for performing a task at hand. For example, document clustering and classification, entity extraction, text summarization, semantic search, and others. How to adequately represent the input text in a machine-interpretable representation that captures its syntactic and semantic structures is still an open research problem.

In this thesis, we identify and address the limitations of existing text representation models. The challenges relate to three major categories: efficiency, effectiveness, and usability of the text representation. We propose new concept-based representations leveraging distributed representations and existing knowledge bases in order to address those challenges. Existing models such as the Bag-of-Words (BoW) and the bag of n-grams suffer from many drawbacks such as: 1) sparsity and the curse of dimensionality impacting their space and computational efficiency, and 2) vocabulary mismatch and lack of word order impacting their effectiveness. Distributional semantics models which represent words as numerical vectors are more efficient but uninterpretable. Explicit concept space models which represent text as Bag-of-Concepts (BoC) are easy to understand and interact with but sparse and suffer from concept mismatch.

Our objective in this thesis is to improve the analysis and retrieval of textual data

especially technical texts (e.g., patents, scientific literature...etc) using the proposed concept-based representations. We show through empirical evaluation that: 1) significant performance improvements can be achieved using our representations with both long technical text (patents) and short text (search queries), 2) our concept-based representations greatly facilitate interactive and visual analysis of technical text, and 3) the proposed conceptual representations are generic and applicable to many academic benchmark datasets where we achieve superior state-of-the-art performance.

First, we present a simple and efficient knowledge-based technique for reducing the dimensionality of the bag of n-grams model. Using our unsupervised technique on a benchmark dataset for patent classification, we achieve 13-fold reduction in the number of bigram features and 1.7% increase in classification accuracy over the BoW baseline.

Second, we address the challenge of short text representation, especially search queries which lack context, order, and syntax (e.g., "software engineer google", "google software engineer"). We propose a novel and effective representation to create an ensemble of contextual, knowledge-based, and lexical features for the given short text. We report the performance of this ensemble representation on entity type recognition of search queries in the recruitment domain. The results show superior performance of our approach over traditional BoW and word embedding models where we achieve 97% micro-averaged F1 score.

Third, we present Mined Semantic Analysis (MSA), a novel concept-based representation model which utilizes unsupervised data mining techniques in order to discover concept-concept associations. These associations are used subsequently to enrich the

BoC representation of the given text. Quantitative evaluation of MSA on benchmark datasets for measuring text semantic similarity shows its superior performance. Additionally, we demonstrate the usability of MSA representations by implementing a Web-based semantic-driven visual and interactive framework for innovation and patent analytics.

Fourth, we propose a neural-based model to learn distributed representations (embeddings) of concepts and entities from their mentions in encyclopedic knowledge bases (e.g., Wikipedia). There are many advantages of this model over sparse representations (i.e., BoW and BoC). First, it is space and computationally efficient. Second, it is more effective as it helps to overcome the concept mismatch problem; here concepts are matched by comparing their embeddings rather than traditional string matching. Third, it is expressive and interpretable. To enhance the learned concept embeddings, we further extend this model by combining the textual knowledge of Wikipedia with the knowledge from Microsoft knowledge graph (Probase). We empirically evaluate the efficacy of the learned representations on benchmark datasets for measuring entity semantic relatedness, analogical reasoning, concept categorization, argument type identification for semantic parsing, and dataless classification where we achieve state-of-the-art performance.

Finally, we address the problem of usability of the text representation. We propose a novel interactive framework for patent retrieval; a domain specific text retrieval task. The proposed framework leverages distributed representations of concepts and entities extracted from the patents text. We also propose a simple and practical interactive relevance feedback mechanism where the user is asked to annotate relevant/irrelevant

results from the top n hits. We then use this feedback for query reformulation and term weighting where weights are assigned based on how good each term is at discriminating the relevant vs. irrelevant candidates. First, we demonstrate the efficacy of the distributed representations on the CLEF-IP 2010 dataset where we achieve significant improvement of 4.6% in recall over the keyword search baseline. Second, we simulate interactivity to demonstrate the efficacy of the proposed interactive term weighting scheme. Simulation results show that we can achieve extra 1.9% to 11.6% improvement in mean average precision from one interaction iteration outperforming previous semantic and interactive patent retrieval methods.

ACKNOWLEDGMENTS

First and foremost, I express my deep and sincere gratitude to Almighty Allah, my God, who gave me the strength and ease during my PhD studies at UNC Charlotte.

I would like to sincerely thank my advisor Wlodek Zadrozny who gave me unlimited support and help during my PhD journey. He always made himself available whenever I sought advice and guidance. I learned a lot from his professional character and his insightful feedback.

I would like to thank the VisCenter, Text Analytics Lab, and the Department of Computer Science at UNC Charlotte for funding my research and providing my graduate assistantship over the course of my PhD.

I would like to thank my dissertation committee members for their constructive feedback about my thesis work. I was honored to have Srinivas Akella, Jing Yang, Zbigniew Ras, and M. Yasin Raja in my committee.

I am very thankful to my colleagues at the Text Analytics Lab for their cooperation, useful discussions, and more importantly their constant friendly spirit.

I am so grateful to Khalifeh Al Jadda, Mohammed Korayem, and Trey Grainger from CareerBuilder for their cooperation and contributions to my project on "Entity Type Recognition for Search Query Understanding" during my internship at CareerBuilder.

I would like to thank Hongxia Jin and her group at Samsung Research America (SRA) for their feedback and fruitful discussions about my project on "Learning Concept and Entity Representations" during my internship at SRA.

I am so appreciative to my family in Egypt for their continuous support and limitless prayers for me. May Allah bless you all.

Finally, I am expressing my deep appreciation and gratitude to my beloved wife Heba Alghor and my darling twins Adam and Lujyne. Thank you Heba! Throughout this long journey, you were strong, understanding, dependable, and nurturing. Without you I never could have achieved this accomplishment.

TABLE OF CONTENTS

LIST OF FIGURES	xvi
LIST OF TABLES	xviii
CHAPTER 1: INTRODUCTION	1
1.1. Text Representation Models	3
1.1.1. Bag-of-Words (BoW)	3
1.1.2. Distributional Semantics	4
1.1.3. Text Conceptualization	5
1.1.4. Knowledge-based Conceptualization	6
1.1.5. Implicit vs. Explicit Representations	7
1.1.6. Brittleness of Syntactic/Semantic Parsing	8
1.2. Thesis Focus	10
1.2.1. Research Questions	11
1.2.2. Hypotheses	13
1.3. Thesis Structure	16
CHAPTER 2: KNOWLEDGE BASED DIMENSIONALITY REDUC- TION FOR PATENT CLASSIFICATION	19
2.1. Background and Related Work	20
2.1.1. Dimensionality Reduction	22
2.1.2. Knowledge Bases in Dimensionality Reduction	23
2.1.3. Patent Classification	23
2.2. Methodology	24
2.3. Dataset and Preprocessing	25

2.4. Experimental Setup	29
2.5. Results	30
2.6. Conclusion	32
CHAPTER 3: ENTITY TYPE RECOGNITION USING AN ENSEMBLE OF DISTRIBUTIONAL SEMANTIC MODELS TO ENHANCE QUERY UNDERSTANDING	34
3.1. Motivation	35
3.2. Related Work	37
3.3. Methodology	39
3.3.1. System Overview	40
3.3.2. The Entity Type Recognition Process	41
3.3.3. Constructing Contextual Vectors	42
3.3.4. Constructing Synonyms Vectors	44
3.3.5. Entity Ontological Features	45
3.3.6. Entity Linguistic Features	46
3.3.7. Building the Prediction Model	47
3.4. Experiments and Results	48
3.4.1. Dataset	48
3.4.2. Experimental Setup	49
3.4.3. Results	51
3.5. Conclusion	54

CHAPTER 4: MINED SEMANTIC ANALYSIS	56
4.1. Text Conceptualization	56
4.1.1. Text Conceptualization Methods	61
4.1.2. Vector-based Concept Space Models	61
4.2. Mined Semantic Analysis	63
4.2.1. The Search Index	64
4.2.2. Association Rules Mining	65
4.2.3. Constructing the Concept Vector	67
4.2.4. Concept Weighting	68
4.2.5. Relatedness Scoring	69
4.3. Experiments on Semantic Similarity and Relatedness	69
4.3.1. Lexical Semantic Relatedness	70
4.3.2. Short Text Similarity	76
4.4. A Study on Statistical Significance	78
4.5. Conclusion	81
CHAPTER 5: INNOVATION ANALYTICS USING MINED SEMANTIC ANALYSIS	84
5.1. Background and Motivation	84
5.2. Case Study	87
5.2.1. Technology Exploration and Landscape Analysis	87
5.2.2. Competitive Intelligence	88
5.3. Conclusion	92

CHAPTER 6: LEARNING CONCEPT AND ENTITY EMBEDDINGS	93
6.1. Background & Motivation	94
6.2. Related Work	98
6.2.1. Text Conceptualization	98
6.2.2. Concept/Entity Embeddings	99
6.2.3. Bag-of-Concepts Densification	101
6.3. Learning Concept Embeddings	102
6.3.1. Skip-gram	102
6.3.2. Concept Raw Context Model (CRX)	103
6.3.3. Concept-Concept Context Model (CCX)	104
6.3.4. CRX vs. CCX	105
6.3.5. Training	107
6.3.6. Creating Continuous Bag-of-Concepts (CBoC)	108
6.4. Text Conceptualization Applications	109
6.4.1. Concept/Entity Relatedness	109
6.4.2. Concept Learning	110
6.4.3. Dataless Classification	110
6.4.4. Bootstrapping	112
6.5. Experiments	113
6.5.1. Entity Semantic Relatedness	113
6.5.2. Concept Categorization	116
6.5.3. Dataless Classification	119
6.6. Discussion & Conclusion	124

CHAPTER 7: LEVERAGING LARGE SCALE KNOWLEDGE BASES FOR LEARNING CONCEPT AND ENTITY REPRESENTATIONS	128
7.1. Introduction	129
7.2. Learning Concept Embeddings	131
7.2.1. Learning from the Text	131
7.2.2. Learning from the Concept Graph	132
7.2.3. Data and Model Training	133
7.3. Evaluation	134
7.3.1. Analogical Reasoning	134
7.3.2. Concept Categorization	136
7.3.3. Argument Type Identification: A Case Study	138
7.4. Conclusion & Discussion	143
CHAPTER 8: PATENT RETRIEVAL: A LITERATURE REVIEW	147
8.1. Introduction	147
8.2. Preliminaries	150
8.2.1. Patent Documents and Kind Codes	150
8.2.2. Patent Classification	152
8.2.3. Patent Families	152
8.3. Data & Evaluation Tracks	152
8.3.1. CLEF-IP Collections	152
8.3.2. NTCIR Collections	156
8.3.3. TREC-CHEM Collections	157

8.3.4. Other Sources	158
8.4. Patent Retrieval Tasks	159
8.5. Patent Retrieval Methods	163
8.5.1. Test Collections & Evaluation Measures	164
8.5.2. Query Reformulation (QRE)	165
8.6. Related Topics	184
8.6.1. Patent Quality Assessment	184
8.6.2. Patent Litigation	187
8.6.3. Technology Licensing	189
8.7. Concluding Remarks	190
CHAPTER 9: TOWARD AN INTERACTIVE PATENT RE- TRIEVAL FRAMEWORK BASED ON DISTRIBUTED REPRESENTATIONS	192
9.1. Introduction	193
9.2. Preprocessing and Offline Operations	195
9.2.1. The Search Index	195
9.2.2. Text Conceptualization	195
9.2.3. Learning Distributed Representations	196
9.3. Automated Vector-based Retrieval	198
9.3.1. Vector Generation	199
9.3.2. Candidate Scoring and Reranking	199
9.3.3. Why Concept-based Distributed Representations?	200
9.4. Interactive Relevance Feedback	201

	xv
9.5. Performance Evaluation	202
9.5.1. Experimental Setup	202
9.5.2. Are the Learned Vectors Meaningful?	203
9.5.3. Retrieval Results	204
9.6. Conclusion	208
CHAPTER 10: CONCLUSION AND FUTURE DIRECTIONS	209
REFERENCES	216
APPENDIX A: LIST OF PUBLICATIONS	235

LIST OF FIGURES

FIGURE 1: Example Parse Trees	8
FIGURE 2: Distribution of Claim-1 Length	9
FIGURE 3: Architecture of the Knowledge-based Dimensionality Reduction System	25
FIGURE 4: Frequencies of the CLEF-IP 2010 Training/Test Samples per Patent Class	26
FIGURE 5: Frequencies of the CLEF-IP 2010 Training/Test Samples over Number of Labels per Sample	27
FIGURE 6: Dimensionality vs. Accuracy Tradeoff	31
FIGURE 7: Architecture of the Entity Type Recognition System	41
FIGURE 8: The General Architecture of Wikipedia-based Concept Space Representation Models	57
FIGURE 9: An Example Concept Graph Representation of the Abstract of Shalaby and Zadrozny [178]	59
FIGURE 10: The Architecture of Mined Semantic Analysis (MSA)	62
FIGURE 11: MSA's Correlation Scores on Lee50 Dataset	77
FIGURE 12: Technology Landscape Analysis of <i>Cognitive Analytics</i>	86
FIGURE 13: Concept Graph of Bank of America's 100 Patent Titles	89
FIGURE 14: Concept Graph of Witricity's 10 Patent Titles	91
FIGURE 15: Bag-of-Concepts Densification using Concept Embeddings	101
FIGURE 16: F1 Scores of Fine-grained Dataless Classification	122
FIGURE 17: F1 Scores of Coarse-grained Dataless Classification	123
FIGURE 18: Integrating Knowledge from <i>Wikipedia</i> Text and <i>Probase</i> Concept Graph	130

FIGURE 19: Taxonomy of Patent Retrieval Methods	163
FIGURE 20: Automated Vector-based Retrieval	194
FIGURE 21: Interactive Relevance Feedback for Term Weighting and Scoring	201
FIGURE 22: Classification of Topic Patents using Class and Document Vectors	203
FIGURE 23: Improvements in Recall and Mean Average Precision of Interactive Relevance Feedback with Iterations	207

LIST OF TABLES

TABLE 1: Evaluating Semantic Similarity using Various Representation Models	2
TABLE 2: Similarity Scoring using Exact String Match vs. Concept Embeddings	15
TABLE 3: CLEF-IP 2010 Dataset Statistics	26
TABLE 4: Concept Counts and Reduction Percentages of the Knowledge Sources	28
TABLE 5: CLEF-IP 2010 Classification Results	30
TABLE 6: Example Search Entities and their Context Vectors	43
TABLE 7: Example Search Entities and their Synonyms Vectors	44
TABLE 8: Distribution of Entities over Categories	48
TABLE 9: Performance of the Contextual Vectors ETR Model	51
TABLE 10: Performance of the Representations Ensemble	53
TABLE 11: Example Concept Representation of <i>Computational Linguistics</i>	63
TABLE 12: MSA’s Pearson Correlation Scores on Benchmark Datasets for Measuring Lexical Semantic Similarity	73
TABLE 13: MSA’s Spearman Correlation Scores on Benchmark Datasets for Measuring Lexical Semantic Similarity	74
TABLE 14: MSA’s Spearman Correlation Scores on MEN Dataset	75
TABLE 15: MSA’s Spearman Correlation Scores for Measuring Short Text Similarity	77
TABLE 16: Steiger’s Z Significance Test on the Differences between Spearman Correlations	79
TABLE 17: Bank of America’s Sample Patent Titles	89

TABLE 18: Witricity’s Sample Patent Titles	91
TABLE 19: Similarity Evaluation using Explicit vs. Distributed Concept Vectors	94
TABLE 20: Contexts of Concept-Raw-Contexts vs. Concept-Concept-Contexts	106
TABLE 21: Spearman Correlation Scores for Measuring Entity Semantic Relatedness	114
TABLE 22: Sample of Top-3 Rated Entities from CRX & CCX on Entity Semantic Relatedness	115
TABLE 23: Accuracy of Concept Categorization using CRX & CCX Models	118
TABLE 24: The 20NG Dataset Category Mappings	120
TABLE 25: Evaluation Results of Dataless Document Classification of Fine-grained Classes	120
TABLE 26: Evaluation Results of Dataless Document Classification of Coarse-grained Classes	123
TABLE 27: Top-5 Related Concepts from CRX & CCX Models	125
TABLE 28: Evaluation Results of the Concept Multimodal Embeddings Model on Analogical Reasoning	135
TABLE 29: Evaluation Results of the Concept Multimodal Embeddings Model on Concept Categorization	137
TABLE 30: Example Utterances and their Logical Forms	139
TABLE 31: Evaluation Results of Semantic Parsing with Argument Type Identification	141
TABLE 32: Patent Kind Codes	151
TABLE 33: Scenarios of Patent Prior Art Search	160
TABLE 34: Keyword-based Patent Retrieval Methods	166

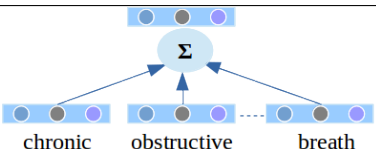
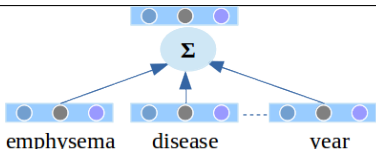
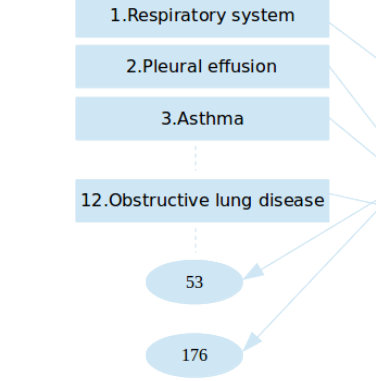
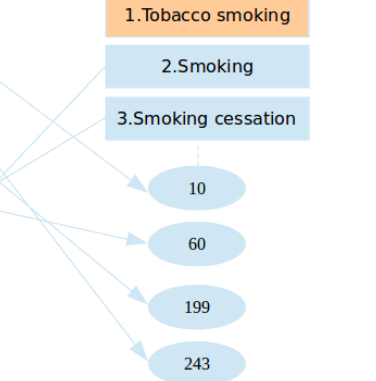
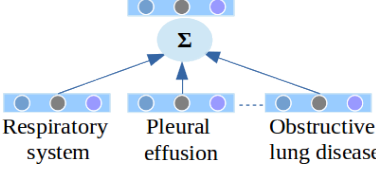
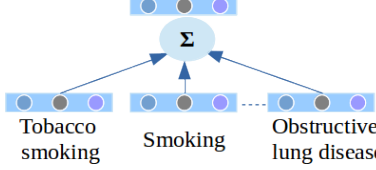
TABLE 35: Pseudo Relevance Feedback Patent Retrieval Methods	169
TABLE 36: Semantic-based Patent Retrieval Methods	174
TABLE 37: Metadata-based Patent Retrieval Methods	178
TABLE 38: Vector-based Patent Retrieval with Interaction	204
TABLE 39: MAP and PRES Scores of Vector-based Patent Retrieval	205
TABLE 40: Percent Improvements After the 1 st Interaction Iteration	205
TABLE 41: Evaluation Results using the Interactive Relevance Feedback Mechanism of Golestan Far et al. [62]	206

CHAPTER 1: INTRODUCTION

Text analytics have emerged recently as a subfield of natural language processing (NLP) to address practical and business problems that involve large and complex collections of textual data. Text analytics represent a set of scalable techniques that mine unstructured and semi-structured textual resources in order to extract useful knowledge, which can be used, directly or indirectly, for performing the task at hand. For example, document clustering and classification, entity extraction, text summarization, semantic search, and others.

One of the major research themes in NLP is concerned with developing representation models of textual data. The main objective of such models is to transform the input text into a machine-interpretable representation. For long texts (e.g., news posts, scientific articles...etc), language models are utilized in order to capture both the syntactic and semantic regularities in textual structures (words, phrases, and document). For short texts (e.g., search queries, tweets...etc), knowledge bases (KBs) are utilized in order to enrich the short contextless text with contextual features and real-world knowledge required to "understand" it.

Table 1: Evaluating semantic similarity between two highly similar text snippets using various representation models. The BoW (A) is the least successful, while the concept space representation (C) and distributed representations (B and D) are relatively better at capturing the semantic similarities.

No	Model	Snippet#1	Snippet#2	Sim
		Chronic obstructive pulmonary disease is an incurable, progressive lung disease that primarily affects smokers and causes shortness of breath and difficulty breathing	Emphysema is a disease largely associated with smoking and strikes about 2 million Americans each year	
A.	BoW	chronic obstruct pulmonary diseas incur progress lung primarili affect smoker caus short breath difficulti	emphysema diseas larg associ smoke strike million american year	0.09
B.	Word2Vec			0.88
C.	BoC			0.81
D.	Continuous BoC			0.91

★ *Emphysema* is a different name to *Chronic obstructive pulmonary disease*.

1.1 Text Representation Models

1.1.1 Bag-of-Words (BoW)

The BoW model (aka the *unigram* model) is the simplest language model. It is a *one-hot*¹ representation under which textual structures are represented as a vector of lexical tokens with frequencies disregarding their order, syntactic structures, and semantics. As we can notice in Table 1-A, the BoW representation is vulnerable to the *vocabulary mismatch* problem, where semantically similar texts would have very low similarity score if they use different vocabulary.

One possible solution to some linguistic limitations of the BoW representation is to utilize the *bag of n-grams* representation where the BoW vector is extended to include tokens of arbitrary length n (e.g., bigrams if $n=2$, trigrams if $n=3$, and so on). This representation though, partially, captures some linguistic patterns (e.g., word order and local compositionality), it suffers from two major drawbacks. First, *the curse of dimensionality* where the number of possible n -gram sequences grows *exponentially* with n . Second, *sparsity* where only a small number of dimensions in the vector will have non-zero frequencies for a target textual structure. These two problems add extra *space* and *computational complexities* for applications utilizing such high dimensional and sparse representation vectors. In addition, similar to the BoW, the bag of n -gram representation still suffers from the vocabulary mismatch problem as it uses one-hot vectors.

¹The one-hot encoding represents each word as a binary vector whose size is equal to the vocabulary size. All the entries are set to 0 except the entry representing the word is set to 1. The BoW representation of a document, is the sum of all the one-hot vectors of the document's words.

1.1.2 Distributional Semantics

Several other representation paradigms have evolved over decades of research in NLP in order to overcome some of the inherent linguistic and computational limitations of the n -gram model. One of the most prominent research areas in language understanding is *distributional semantics*. These models are inspired by the distributional hypothesis [75] which emphasizes the idea that similar words tend to appear in similar contexts and thus have similar contextual distributions. Therefore the meaning of words can be determined by analyzing the statistical patterns of word usage in various contexts over large textual corpora.

Distributional semantics methods are either corpus-based or lexicon-based depending on the resource from which world knowledge is acquired and used to represent the input text as "meaning" vectors. *Corpus-based methods* utilize large textual corpora to analyze *local word contexts* creating co-occurrence statistics between words. Then these statistics are used to generate *implicit representations* as low dimensional real-valued vectors of words (aka *embeddings*, *distributed*, *continuous* or *dense* vectors). Examples of these methods include LSA [105], LDA [16], and more recently neural-based embeddings such as Word2Vec [138] and Glove [154]. On the other hand, *lexicon-based methods* utilize explicit word relations found in human-built dictionaries such as Wordnet [49] and Wiktionary² in order to determine word meanings [22, 218, 155]. The promise of such methods is that: syntactic and semantic features of words would be encoded in the produced vectors.

²<https://www.wiktionary.org>

Table 1-B shows how word embeddings from the Word2Vec model³ could overcome the vocabulary mismatch problem. As we can notice, by averaging the word vectors of each text snippet and comparing the resulting average vectors, we obtain a more representative similarity score of 0.88 compared to the BoW score of 0.09. It is worth mentioning that, less relevant words to the meaning of the given text (e.g., primarily, largely, million, year...etc) would still contribute to the average vectors, and thus might add noise to the final representation of the given text.

1.1.3 Text Conceptualization

Another active line of research is concerned with *explicit semantic representations* of texts as *bag-of-concepts* through text conceptualization. Such methods focus on the global contexts of terms (i.e., documents in which they appeared), or their properties in existing KBs in order to figure out their meanings. Text conceptualization is motivated by the fact that humans understand languages through multi-step cognitive processes which involve building rich models of the world and making multilevel generalizations from the input text [209]. One way of automating such generalizations is through text conceptualization. Either by extracting basic level concepts and entities from the input text using concept KBs [96, 184], or mapping the whole input into a concept space that captures its semantics as in ESA [56] and MSA (described in Chapter 4).

³Word embeddings were obtained by training Word2Vec on Wikipedia. And we use 500 dimensional word vectors.

1.1.4 Knowledge-based Conceptualization

One major category of conceptualization methods utilizes semi-structured KBs such as *Wikipedia* in order to construct the concept space which is defined by all *Wikipedia article titles*. Such models have proven efficacy for semantic analysis of textual data especially short texts where contextual information is missing or insufficient. For example, measuring semantic similarity/relatedness [56], dataless classification [31, 181, 182, 110], search and relevancy ranking [45], event detection and coreference resolution [153].

Another category of conceptualization methods utilizes more structured concept KBs such as *Microsoft Knowledge Graph* (aka. *Probase*⁴) [212]. *Probase* is a probabilistic KB of millions concepts and their relationships (basically is-a). It was created by mining billions of Web pages and search logs of Microsoft’s Bing⁵ repository using syntactic patterns. The concept KB was then used for text conceptualization to support text understanding tasks such as clustering of Twitter messages and News titles [183, 184], search query understanding [210], short text segmentation [210, 86], and measuring term similarity [96, 109].

Table 1-C shows the Bag-of-Concepts (BoC) representation of both text snippets using MSA⁶ (described in Chapter 4). As we can notice, the main focus of snippet#1 is on the disease and its impact on breathing, which is pretty well reflected in the top concepts. On the other hand, snippet#2 is focused on the disease and smoking

⁴<https://concept.research.microsoft.com>

⁵<https://www.bing.com/>

⁶MSA uses *Wikipedia* as the concept space repository, and we create 500 dimensional BoC vectors for each snippet.

which is captured at the top concepts as well. The arrows in Table 1-C, indicate the position each concept in one BoC appeared on the other BoC. Out of 500, we found 109 common concepts between the two BoC vectors, but they appear at different ranks. The final similarity score is relatively representative (>0.80). However, it is less than the Word2Vec model score. One reason for that is the *sparseness* of the BoC where each dimension represents a concept and thus the similarity depends on *exact matching* between corresponding dimensions. This prevent the scoring function from matching different but similar concepts. For example, "*Tobacco smoking*" appeared on one side but not the other so its contribution to the similarity is zero though it has high similarity to "*Smoking*" on the other side. The BoC has an upper hand when it comes to *understandability* and *expressiveness* of its dimensions (concepts). The dimensions of word vectors on the other hand are *not readable*.

1.1.5 Implicit vs. Explicit Representations

As pointed out by Wang and Wang [209], *implicit representations* are dense and thus more computationally efficient. However, these representations are just a bunch of real-valued numbers, and therefore not human friendly. Besides, representations of rare and new words are either poor or missing. *Explicit representations*, on the other hand, can be easily understood by humans and thus easier to interact with. However, the concept space is usually very huge, resulting in very large model. Besides, the BoC suffers from data sparsity causing distant representations for vectors containing concepts with similar meanings in the concept space.

As we will show in Chapters 6 and Chapter 7, we can overcome the *concept mis-*

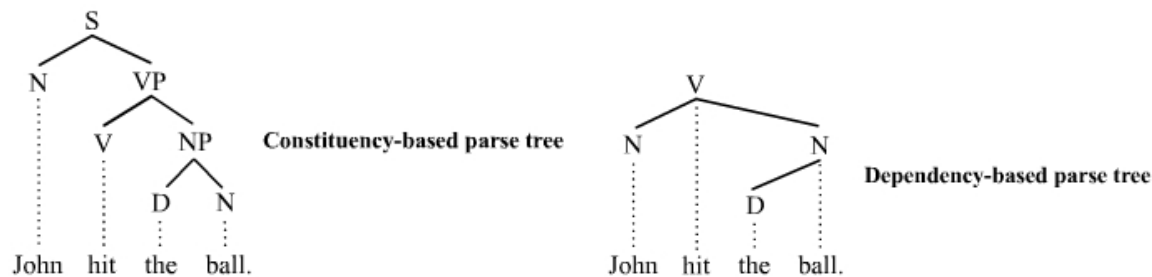


Figure 1: Example parse trees for "John hit the ball".

match problem by learning distributed representations of concepts (concept embeddings). By learning such embeddings, it would be easy to compare pairs of concepts using their embeddings rather than string matching. Table 1-D shows how the same sparse BoC vectors in row C could be better matched by first performing weighted average the individual concept vectors to get corresponding continuous BoC vector (CBoC), and then comparing the average dense vectors to get a more representative similarity score of 0.91 between the two snippets. Under such CBoC representation, we make sure that: 1) each concept in the BoC would proportionally contribute to the final meaning of the given text according to its importance⁷, and 2) each concept would proportionally contribute to the overall similarity score even if it appears in one BoC but not the other (e.g., "Tobacco smoking").

1.1.6 Brittleness of Syntactic/Semantic Parsing

Natural Language Understanding is an *AI complete* problem. Building a computer program that fully understands the text requires modeling and reasoning about various types of *world and commonsense knowledge* which appeared to be very challenging task.

⁷We will describe different concept weighting functions in Chapter 4. Typically, importance is proportional to the concept rank in the BoC.

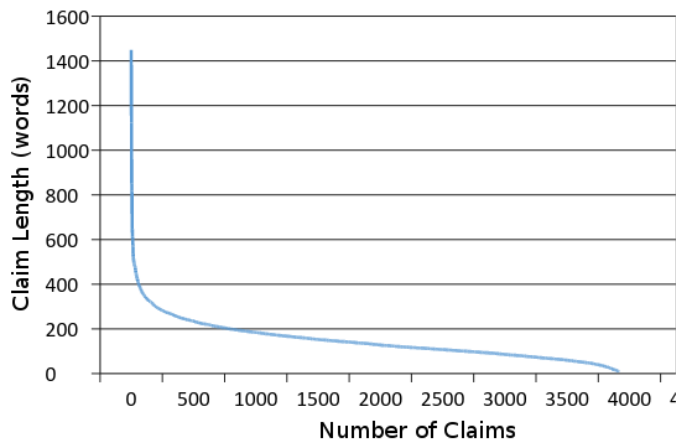


Figure 2: Distribution of Claim-1 length in 4000 patents related to sustainability (from Rajshekhar et al. [161]).

Current methods to language understanding are limited to simple and/or restricted tasks. And they try to *approximate human-like behavior* when it comes to text understanding. For example, syntactic parsing appears to be an enabler to language understanding by helping us to understand *who* did *what* to *whom*. Consider the parse trees of "John hit the ball" in Figure 1. It is easy having these parse trees to answer questions like *Who hit the ball?*, *What John hit?...*etc. by following the tokens part-of-speech in the constituency tree (left⁸), or the branches of the dependency parse tree (right⁹). However, syntactic and dependency parsing become *brittle* when dealing *with very long or very short sentences*.

Regarding long text, Rajshekhar et al. [161] highlighted that, "*The average sentence length in the Wall Street Journal corpus is 19.3 words, ranging from 3 to 20 [189]. And most natural language parsers are trained on similar corpora. In contrast, when we look at the US patent corpus, we can find that average patent claims length is longer.*

⁸https://commons.wikimedia.org/wiki/File:Parse_tree_1.jpg

⁹<https://commons.wikimedia.org/wiki/File:Parse2.jpg>

For example, on a sample of about 4000 patent related to resilience and sustainability, the average length of Claim 1 is 147.3 words, ranging from 7 to 1449, with a standard deviation of 91 words, as depicted in Figure 2. Note that 93% of the claims in this series are longer than 50 words. We get similar results looking at a week of granted patents (5200 patents from 2015) where we get the average Claim 1 length to be close to 190 words. Prior research in this area shows how parsing accuracy decreases with the length of the sentence. For example, McDonald and Nivre [136] shows a parsing accuracy drop of 10 points or more per 40 words. This means that an analysis of the structure for an average Claim 1 is likely to be wrong. An analysis of parsing of sentences up to the length of 156 by Boullier and Sagot [19] entertain a possibility that "(full) parsing of long sentences would be intractable".

Brittleness of syntactic parsing appears with short texts as well especially search queries where syntactic compositionality does not exist most often (e.g., *no word order, no function words, no context...etc*). For example, it is quite common with general purpose search engines that users may use same words in different order to express the same information need (e.g., *software engineer google* and *google software engineer*). Under these circumstances, the need for concept-based and entity-based knowledge bases appears to be inevitable in order to *enrich* our knowledge about search query entities.

1.2 Thesis Focus

In this thesis, we focus on semantic representations of textual content with special attention to *concept-based representation models*. We study the characteristics of

these models in an attempt to understand and identify the limitations and challenges associated with them. Understanding such challenges allows us to further propose adequate and novel solutions to unleash the powers of such models for superior text mining and retrieval performance. In this section we present these challenges, the research questions to be addressed, and the hypotheses to be tested throughout the thesis.

1.2.1 Research Questions

The semantic representation challenges addressed in this thesis relate to three major categories: *efficiency* of the representation, *effectiveness* of the representation, and *usability* of the representation.

1. Efficiency of the representation: It refers to the space and computational complexities to process text under specific representation. Here we address:

- What are the characteristics that make the text representation model more/less computationally efficient?
- How does the dimensionality of the representation impact its performance and computational efficiency?
- Can relevant dimensions (features) be acquired from existing KBs, and what is cost/benefit of this approach with respect to the performance and efficiency?
- What are the computational challenges of existing concept-based representations?

- How can we utilize existing KBs and representation learning to improve the computational efficiency of the sparse high-dimensional bag-of-concepts models?

2. Effectiveness of the representation: It refers to the quality of results when processing text under specific representation. Here we address:

- What is the relation between the dimensionality of the concept-based representation and its performance, and how can we quantify such relation?
- How does sparsity affect the performance of the concept-based representation?
- Can we use existing KBs to increase the effectiveness of existing concept-based representations? For example, augmenting the bag-of-concepts vector with more related concepts without supervision.
- How can we exploit existing KBs and distributed representations to more effectively represent and understand both technical text (e.g. patents) and short text (e.g., search queries)?
- Which is more effective, distributed or discrete concept vectors?

3. Usability of the representation: It refers to the ability to visualize, understand, and interact with the specific representation. Here we address:

- How can we combine the efficiency of the distributed representations with the interpretability of the concept-based representations?

- What visual and interactive techniques can be employed to give non-experts easy and effective ways to work with the concept-based representations?
- How can we evaluate the effectiveness of such interactive frameworks in real-life applications using user-centered methods (e.g., semantic search using concept-based representations and interactive query reformulation)?

1.2.2 Hypotheses

We test several hypotheses in this thesis in order to answer all the aforementioned research questions.

H1. Existing KBs contain rich and huge amount of both general and domain specific knowledge. It is hypothesized that *concepts and entities* in such KBs along with their relationships could be used to *enrich* the semantic representation of textual structures (e.g. technical text) and subsequently improve the retrieval performance of such structures.

For example, the multiword concepts capture local compositionality and therefore could be used as relevant low cost n-gram features to enrich the bag-of-words representation. In addition, *text conceptualization* through mapping into the concept space or identifying basic level concepts could be a *prototype* for the human cognitive process of *generalization*, and thus help better capture the basic ideas and characteristics of the input text. Consider for example the below definition:

”Chronic obstructive pulmonary disease is an incurable, progressive lung disease that primarily affects tobacco smokers and causes shortness of breath and difficulty breathing.”

And let’s assume we perform text conceptualization using *Wikipedia* as the underlying source of concepts. Here we can easily get all the bold multiword expressions corresponding to *Wikipedia* articles (concepts). As we can notice:

- 1) these concepts are capturing relevant information about the definition, and
- 2) we can easily obtain up to 4-gram expressions with just intersecting the our vocabulary with all *Wikipedia* titles. If we want to obtain such relevant features with the bag of n-grams, we will have to also include some irrelevant and noisy features such as the underlined multiword expressions. Moreover, the vector size will be much larger leading to unnecessary space and computational complexities when processing such vector.

H2. Distributed representations are *fixed length* vectors (typically few hundreds). Therefore they are more *space and computationally efficient* than the sparse high dimensional bag-of-concepts representations. The bag-of-concepts, on the other hand, is more *expressive and easy to interact with*. It is hypothesized that we can combine the benefits of both worlds as follows: First, we learn robust distributed representations of concepts which are the basic building blocks of the bag-of-concepts. Second, we use these dense concept vectors to generate fully continuous bag-of-concepts. Finally, we employ the original bag-of-concepts as the presentation layer, and the continuous bag-of-concepts as the computation

Table 2: Keywords representing *Motorcycle* category and a sample from the 20-newsgroups dataset. Top 5 concepts for each are generated using ESA. Using exact match similarity scoring (ESA) result in 0.0 score as no common concepts exist. Dense BoC gives higher and more representative similarity score of 0.69.

Category (Motorcycle) (keywords + top 5 concepts)	Sample (Motorcycle) (text + top 5 concepts)	ESA	Dense BoC
<p>"bike motorcycle yamaha"</p> <ul style="list-style-type: none"> - Outline of motorcycles and motorcycling, - Yamaha YZ450F, - Yamaha Motor Company, - Motorcycle, - 2002 Grand Prix motorcycle racing... 	<p>"is it possible to do a wheelie on a motorcycle with shaft drive as the owner of a v sabre shaftie i can answer from personal experience aieeeeeeeeeeeeeee chuck smythe dod 0 re shaft drives and wheelies"</p> <ul style="list-style-type: none"> - Evel Knievel, - Wankel engine, - History of BMW motorcycles, - Traxxas, - Gas turbine, 	0.0	0.69

layer. In this way we can combine computational efficiency of distributed representations with expressiveness and usability of conceptual representations in *one two-sided concept-based representation*.

H3. Current explicit concept representation models use *exact string matching* in order to measure the similarity between pairs of bag-of-concepts. This requires creating a sparse vector with a few hundred concepts in the first place hoping to have sufficient number of common concepts between the given pair. Creating such vectors is typically *costly* (e.g., in case of ESA [56], it requires searching an index of millions of articles and retrieving the top n hundreds). Dense bag-of-concepts vectors allow us to match concepts using their *embeddings* and hence it guarantees *non-zero similarity* score between different but related concepts. It is hypothesized that using such vectors will allow us to operate more efficiently and effectively *with less number of dimensions* in the vector space. Therefore,

we can reduce the cost of creating the bag-of-concepts as we will retrieve less concepts rather than few hundreds. In addition, presenting less number of concepts to users will reduce the *cognitive load* required to interact with them, thus improving the usability.

Consider the example in Table 2. As we can notice, if the bag-of-concepts pair has no overlapping concepts. Therefore, the exact match similarity scoring gives 0.0 similarity. With dense bag-of-concepts, we can overcome this mismatch problem as similar concepts will have similar embeddings and hence we get a higher and more representative similarity score of 0.69.

1.3 Thesis Structure

The remainder of this thesis is organized as follows:

In Chapter 2 we focus on the increasing the efficiency of the representation through dimensionality reduction especially technical text representation. We report a low-cost approach using knowledge-based concepts for reducing the dimensionality of the bag of n-grams model, while maintaining competitive performance. We evaluate the performance of the knowledge-based model on patent classification and show its low computational and space costs, and its effectiveness on that task.

In Chapter 3 we focus on the increasing the effectiveness of the representation especially short text representation. We introduce our work on short text understanding leveraging existing KBs and distributed representations to create an ensemble which captures contextual, knowledge-based, and lexical features of the given short text. We report the performance of this ensemble representation on entity type recognition

of search queries and show its superior performance over traditional bag-of-words and word embedding models.

In Chapter 4 we focus on the increasing the effectiveness and usability of the bag-of-concepts. We describe a novel concept space representation model which we name Mined Semantic Analysis (MSA). MSA employs encyclopedic KBs and data mining techniques in order to learn concept-concept associations. Thereafter, these associations are used to enhance the expressiveness of the bag-of-concepts representation. We report the performance of MSA on measuring the semantic relatedness of words and sentences and show its effectiveness over other concept-based and vector-based representation models.

In Chapter 5 we present a case study on implementing a concept-based visual and interactive framework powered by MSA for innovations and patents analytics. We demonstrate applying the acquired knowledge from MSA representations to support many cognition and knowledge-based use cases for innovation analysis including technology exploration and landscaping, competitive analysis, prior art search and others.

In Chapter 6 and Chapter 7, we focus on increasing the efficiency and effectiveness of the interpretable bag-of-concepts models. We present our work on learning concept embeddings utilizing neural networks and large-scale KBs. We also propose an efficient low cost mechanism for bag-of-concept densification using the learned embeddings. Through empirical results, we demonstrate the effectiveness of these embeddings in various tasks including: 1) measuring entity semantic relatedness and ranking, 2) concept categorization, 3) dataless text classification using continuous BoC vectors, and 4) analogical reasoning. Additionally, we present a case study to

extrinsically evaluate the learned embeddings on unsupervised argument type identification for neural semantic parsing where we achieve competitive performance with the ability to better generalize to rare and out of vocabulary concept and entity mentions.

In Chapter 8 and Chapter 9, we focus on increasing the effectiveness and usability of the representation especially technical text representation. We address a very challenging text retrieval task; patent prior art search. We start with a literature review on patent retrieval in Chapter 8. Then, we introduce a novel interactive framework for patent retrieval leveraging: 1) distributed representations of concepts and entities extracted from the patents text, and 2) a simple practical relevance feedback interaction mechanism. We show the efficacy of the proposed framework through empirical evaluation on a benchmark dataset for patent search where we outperform previous semantic and interactive patent retrieval methods.

Chapter 10 concludes the thesis, and highlights our contributions and future work.

CHAPTER 2: KNOWLEDGE BASED DIMENSIONALITY REDUCTION FOR PATENT CLASSIFICATION

In this chapter we focus on the increasing the efficiency of the representation especially technical text. We address the curse of dimensionality problem associated with the bag of n-grams representation. We propose a novel and simple technique for dimensionality reduction using concept mentions in freely available online Knowledge Bases (KBs) to improve technical text retrieval. The complexity of this method is linearly proportional to the size of the full feature set, making it applicable efficiently to huge and complex datasets. We demonstrate the effectiveness of our approach on patent data, the largest free technical text. We report empirical results on classification of the CLEF-IP 2010 dataset using bigram features supported by mentions in encyclopedic KBs (*Wikipedia*), lexical KBs (*Wiktionary*), and statistical lexicons (*GoogleBooks*¹⁰). Using our unsupervised method, we achieve 13-fold reduction in the number of bigrams features and 1.7% increase in classification accuracy over the bag-of-words baseline. Though this accuracy score is not the best reported on this dataset, it demonstrates that concept-based representations have a potential for increasing technical text retrieval performance. In addition, these results give concrete evidence that massive reduction in dimensionality and accuracy improvements could be achieved using our approach alleviating the tradeoff between the representation ef-

¹⁰<http://storage.googleapis.com/books/ngrams/books/datasetsv2.html>

efficiency and effectiveness. The results also validate our hypothesis that KB concepts could be used as relevant low cost n -gram features to increase the effectiveness of the bag-of-words representation without greatly degrade its space and computational efficiency.

2.1 Background and Related Work

When it comes to text mining, space and computational complexities arise as most applications involve high dimensional and sparse feature vectors. These complexities even increase when extending the Bag-of-Words (BoW) model commonly used for document representation to include (in addition to words/unigrams) linguistic phrases such as bigrams and trigrams. Despite this additional complexity, these features proved their significance in different text mining tasks especially text classification [190, 43]. Therefore, a *tradeoff* between representation *efficiency* (in terms of space and computation) and *effectiveness* (in terms of accuracy) that needs to be resolved.

For text classification, many *dimensionality reduction* techniques were used to resolve this tradeoff [104, 215, 16, 215]. These techniques aim to reduce the number of features (thus increasing efficiency) while keeping most of the variance in the data (thus maximizing accuracy).

Simple approaches like Term Frequency (TF), Document Frequency (DF), Category Frequency Document Frequency (CF-DF), and Term Frequency Inverse Document Frequency (TF-IDF) work by first assigning a relevancy score based on feature counts in the text corpus, and then pruning all those features whose score is under specific threshold. Other *supervised* approaches inspired by information theory like

Information Gain (IG), Mutual Information (MI)¹¹, and Chi-square (χ^2) statistic¹² work similarly [134]. Another set of approaches like Principal Components Analysis (PCA) and Independent Component Analysis (ICA) are *unsupervised*. They work by finding a reduced set of dimensions on which to project the data such that most variation of the data is maintained, and then use them to create new low dimensional feature vectors for the given task. We can notice two main disadvantages of these transformation methods: 1) they are computationally costly, and 2) the generated features set cannot be interpreted easily by humans.

Automated patent classification represents a concrete example of the curse of dimensionality problem associated with *large text documents*. Patent data represent the largest technical text corpus that is freely available. Therefore, the need for effective dimensionality reduction techniques becomes inevitable for this task. As highlighted by Benzineb and Guyot [15], automated patent classification gets its importance from: 1) the continuous rise in the number of patents applications every year, 2) the need to maintain consistent patent classification as new categories and subcategories emerge, and 3) to serve patent prior art search.

Patent classification assigns a code to each patent document according to a predefined *classification scheme*. This classification code reflects the *technical features* of the patent. Here we focus on the International Patent Classification (IPC) scheme where the classification hierarchy is defined as a tree structure [15]. The *Section* level

¹¹MI measures how much information the existence/absence of the feature contributes to predicting the class correctly [134]. IG works similarly

¹²The χ^2 statistic tests the independence of two events. In feature selection, one event corresponds to the class and the other corresponds to the feature. Dependence implies the feature is relevant in predicting the class [134].

is the top one and below come descendant levels; *Classes*, then *Sub-Classes*, then *Groups*, and finally *Sub-groups*.

2.1.1 Dimensionality Reduction

Text classification using the BoW representation involves high dimensional feature spaces. As many classification techniques are computationally sensitive to the size of feature vectors, different feature reduction approaches were used to alleviate this problem. Lam and Lee [104] provided a comparative study of four feature selection methods for text classification; namely: document frequency, category frequency document frequency, term frequency inverse document frequency, and Principal Components Analysis (PCA). They found that PCA was the most effective method. Another study by Yang and Pedersen [215] indicated high correlation between document frequency, Information Gain (IG), and chi-square (χ^2) scores though better results were achieved using IG and χ^2 when applied to the Reuters corpus¹³ classification task.

Other data transformation and compression methods were proposed to find low dimensional spaces that best represent textual data. Blei et al. [16] proposed Latent Dirichlet Allocation (LDA) which represents each document as a set of topics with probabilities. LDA with the Reuters dataset achieved a huge vocabulary reduction with almost the same accuracy compared to the BoW representation. Transformation methods like PCA and LDA, though effective, are computationally more costly than our approach. Both PCA and LDA take polynomial time in the number of input features [185], while our approach is linearly proportional to the size of the full feature

¹³<http://www.daviddlewis.com/resources/testcollections/reuters21578/>

set, making it more efficient and scalable to huge datasets.

2.1.2 Knowledge Bases in Dimensionality Reduction

The use of knowledge bases for dimensionality reduction was not reported before. A closely related work by Gabrilovich and Markovitch [55] is the use of *Wikipedia* articles for feature generation to enhance text categorization. In this work, each document BoW is extended to include relevant concepts from *Wikipedia* articles. It's worth noting that this approach aims mainly to expand the corpus vocabulary and thus increasing the dimensionality of the feature space, therefore, it requires applying a feature selection technique afterwards to eliminate the irrelevant features.

Wikipedia articles were used extensively for different text mining tasks like co-reference resolution [164], news visualization [58], and text classification [202, 181]. In the IBM Watson system [34], *Wikipedia* article titles were used for candidate answer generation and for data preparation to create title oriented textual resources.

2.1.3 Patent Classification

The use of statistical phrases as features for text classification has been analyzed in [190, 29, 14], and for patent classification in [43, 66]. Tan et al. [190] explored adding bigrams to the set of unigram features to enhance classification accuracy. To avoid high dimensionality, reduction techniques such as information gain, term frequency, and document frequency was used to keep the most relevant bigrams. The results indicated significant improvements of accuracy using this approach. Caropreso et al. [29] noted that, although methods such as document frequency, information gain, and chi-square assigned higher scores to bigrams than unigrams, bigrams were not effective

for classifying the Reuters dataset . Bekkerman and Allan [14] justified these contradicting results by hypothesizing that the effectiveness of a small number of "good" bigrams is nullified by a huge number of "junk" bigrams. This hypothesis coincides with our objective to develop a robust reduction technique that discriminates "good" vs. "junk" bigrams and hence keeps only "good" ones. In the context of patent classification, D'hondt et al. [43] reported significant improvement in accuracy when using bigrams along with unigrams. Nevertheless, their approach expanded the initial unigrams vocabulary (~ 58 thousand terms) by 20-fold when adding bigrams (~ 1.1 million terms). This indeed represents an example of the tradeoff between high dimensionality and classification improvements associated with statistical phrases used as classification features.

2.2 Methodology

We propose a novel and simple approach for dimensionality reduction by utilizing freely available Knowledge Bases (KBs) as resources of relevant n -gram features, especially *bigrams*. Free online KBs represent sources of different types of knowledge including technical expressions definitions and references in the form of concepts and entities.

As patents language is highly technical and open domain, we utilize three knowledge sources; namely: *Wikipedia* article titles, *Wiktionary* article titles, and *GoogleBooks* bigrams. Our hypothesis is that: relevancy of bigrams extracted from patent documents can be supported by mentions in knowledge sources, therefore, only those bigrams that are referenced in target knowledge source should be kept and all others

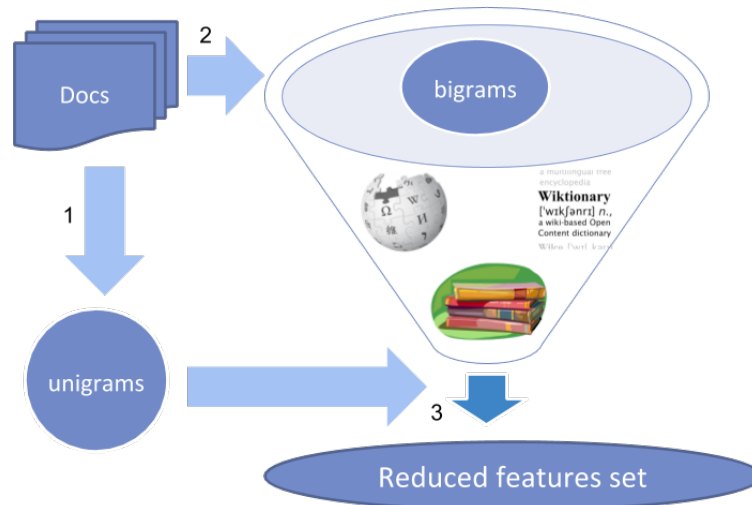


Figure 3: Architecture of our knowledge-based dimensionality reduction system.

pruned.

Figure 3 shows how we obtain the reduced feature set given a collection of documents. First, we generate unigram features. Second, we generate bigram features and intersect them with mentions in the three knowledge sources (*Wikipedia*, *Wiktionary*, and *GoogleBooks*). Finally, we combine both the unigrams and the filtered bigrams to obtain the reduced set of features.

2.3 Dataset and Preprocessing

We used the CLEF-IP 2010 patents collection¹⁴; it contains approximately 2.6 million documents corresponding to 1.3 million individual patents. The documents are in XML format, and they may contain text in different languages including English, German, and French. In our experiments we used approximately 0.5 million English patent *abstracts*.

We started parsing the XML documents considering only ones that end with

¹⁴<http://www.ir-facility.org/collection/>

Table 3: A summary of CLEF-IP 2010 dataset statistics after preprocessing.

	All Samples	Training Samples	Test Samples
# of docs	514,365 (100%)	411,484 (80%)	102,872 (20%)
# of labels	121	121	121
max. labels/doc	12	12	11
min. labels/doc	1	1	1
avg. labels/doc	1.6	1.6	1.6
# of unigrams	65,623	58,661	30,774
# of bigrams	1,261,884	1,073,805	377,338

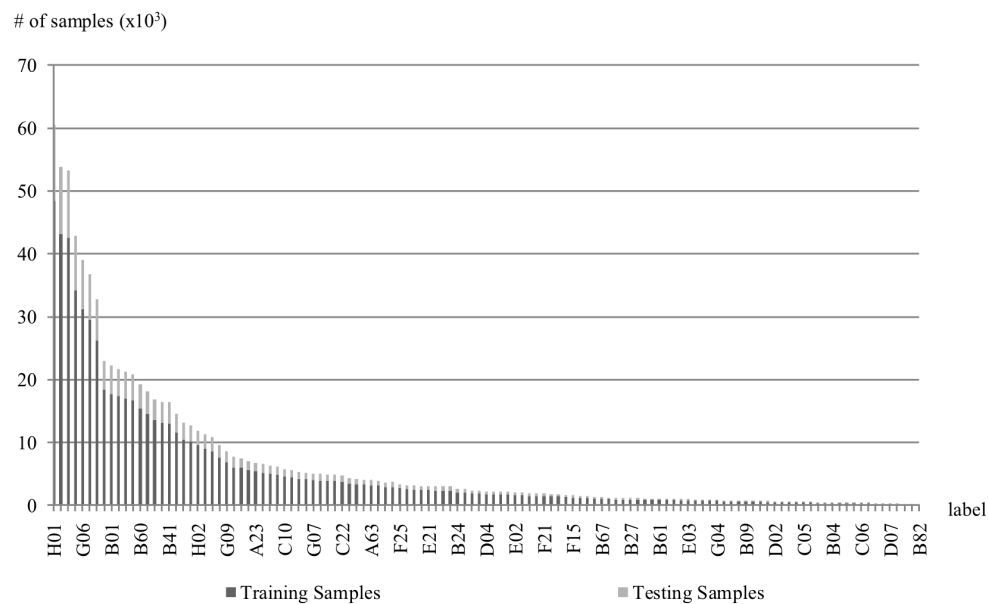


Figure 4: Frequencies of the training/test samples over the 121 class labels. The long tail distribution makes classification of unpopular classes, which have small number of samples, more challenging. For example classes B04 and C06..

”A1.xml” or ”A2.xml” which correspond to patents applications published with/without prior art search report respectively. We considered only documents where *abstract* text was provided in English and *description* text was not missing. All experiments used the patent abstract text as the main source of features. Due to some inconsistencies in the data, there were *abstract* tags attributed as English while the text itself

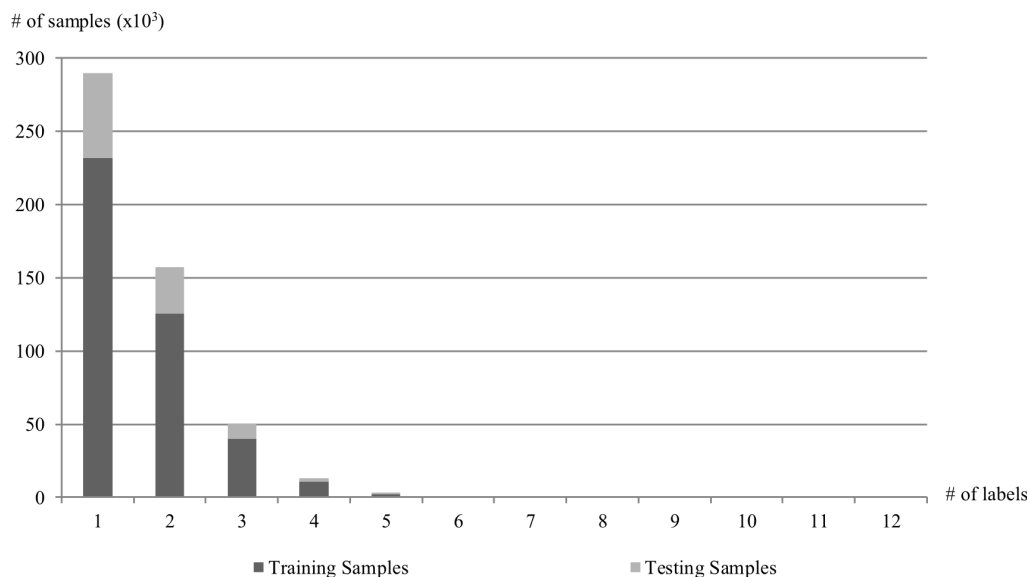


Figure 5: Frequencies of the training/test samples over the number of labels per sample. Almost all samples have at most 4 labels and less than 1% have > 4 labels

was not in English, for this reason, we used a language detection tool ¹⁵ to exclude non-English *abstracts*.

As shown in Table 3, overall XML filtration resulted in 514,356 English *abstracts* which were subsequently divided randomly into two subsets: 1) a training set of 411,484 *abstracts* representing 80% of the data, and 2) a testing set of 102,872 *abstracts* representing 20% of the data. We considered only the codes on class level (1 letter section symbol + 2 digits class code e.g., "H01") which resulted in a total of 121 labels with an average of 1.6 labels per patent document.

Frequencies of the training/testing samples over the 121 labels are shown in Figure 4, the long tail distribution of unpopular labels makes the classification task more challenging. Distribution of training and testing samples over different numbers of

¹⁵<https://github.com/saffsd/langid.py/>

Table 4: Knowledge sources concept counts and reduction percentages after intersecting with training samples bigrams. We demonstrate using bigrams only.

Bigrams Source	# of Bigrams	% of Reduction
Training Set	1,073,805	0%
Wikipedia \cap Training	41,397	96.14%
Wiktionary \cap Training	8,764	99.18%
GoogleBooks \cap Training	54,579	94.92%
(Wikipedia \cup Wiktionary) \cap Training	43,456	95.95%
(Wikipedia \cup GoogleBooks) \cap Training	81,583	92.40%
(Wiktionary \cup GoogleBooks) \cap Training	59,826	94.43%
All \cap Training	83,393	92.23%

labels is shown in Figure 5; almost all samples have at most 4 labels, and less than 1% of them have > 4 labels.

For preprocessing we followed most preprocessing steps in D’hondt et al. [43]. We converted all *abstract* texts into lower case, removed all *claims* and figures references, removed all list references, and removed digits and punctuations characters. We then lemmatized all text tokens. Finally, we ran a simple tokenizer to produce all texts unigrams and bigrams pruning all those whose document frequency < 2 and term frequency < 3 . Overall the unigrams and bigrams counts are shown in Table 3.

For experiments we collected bigrams from each knowledge source and intersected them with the whole collection of bigrams keeping only bigrams that are mentioned in both the corpus and the knowledge source. For *Wikipedia* (version 2014-09-03) and *Wiktionary* (version 2014-09-08), we used all titles of two words, and for *Google-Books* (version 2012-07-01) we repeated same procedure using 2-grams books titles. Thereafter, all bigrams were lemmatized and intersected with lemmatized bigrams from the training set samples. Table 4 shows bigrams counts and reduction percentages of each source as well as all sources combined. As we can notice, our approach

resulted in massive reduction in dimensionality of the training set bigrams features. Even when combining all bigrams from all sources, approximately 13-fold reduction in dimensionality was achieved resulting in immense reduction of the representation space complexity and thus increasing its computational efficiency. On the other hand, our approach is *linearly* proportional to the number of bigrams in the training set, hence very *efficient for huge feature spaces*.

2.4 Experimental Setup

The main goal of the experiments is to investigate the impact of massive dimensionality reduction guided by concepts from KBs on classification accuracy. For this reason, experiments were not tuned toward achieving the best classification performance, but rather to investigate relative gains in accuracy when using KBs concepts (basically bigrams) compared to using the whole set of bigrams. For experiments, we used the scikit-learn machine learning library [152], and TF-IDF for feature vector representation. Classification was then performed using linear SVM classifier which learns a linear support vector classifier. It is implemented using libLinear¹⁶ and scales very well to problems with millions of instances and features. As our task is multi-class and multi-label classification, we used One-vs-Rest scheme where a classifier is built for each class label. Promotion scheme is then followed to determine label(s) of each sample by selecting label(s) with highest membership probabilities. Because linear SVM implementation doesn't support class probabilities, we used the sigmoid function to convert the linear SVM decision function output (z) into a probability (p)

¹⁶<http://www.csie.ntu.edu.tw/~cjlin/liblinear/>

Table 5: Classification results of CLEF-IP 2010 dataset. "Uni" are unigrams, "Bi" are bigrams, "All KB Bi" are bigrams from the three knowledge sources.

No	Features Source	P	R	F1
1	Uni	72.50	66.67	69.46
2	Uni \cup All Bi	75.98	69.59	72.64
3	Uni \cup Wikipedia Bi	73.68	68.13	70.80
4	Uni \cup Wiktionary Bi	72.98	67.21	69.98
5	Uni \cup GoogleBooks	73.46	68.16	70.71
6	Uni \cup Wikipedia Bi \cup Wiktionary Bi	73.67	68.18	70.82
7	Uni \cup Wikipedia Bi \cup GoogleBooks Bi	73.92	68.59	71.15
8	Uni \cup Wiktionary Bi \cup GoogleBooks Bi	73.59	68.30	70.84
9	Uni \cup All KB Bi	<u>73.94</u>	<u>68.63</u>	<u>71.19</u>

as follows:

$$p = \frac{1}{(1 + e^{-z})} \quad (1)$$

After trying different values, we chose a threshold of 0.45 for class membership where each sample is assigned to label(s) above the threshold with a maximum of 4 labels per sample. We also configured the classifier to assign a minimum of one label per sample even if its probability is under the threshold. We report classification accuracy by measuring Precision (P), Recall (R), and $F1$ measure micro-averaged on document level. For $F1$ calculation we used the weighted average of P and R as follows:

$$F1 = \frac{2PR}{P + R} \quad (2)$$

2.5 Results

Table 5 shows classification results of 9 experiments conducted using unigrams, bigrams, and combinations of both unigrams and bigrams from the three knowledge sources. These experiments were intended to measure the relative improvement in accuracy vs. previously reported dimensionality reduction in Table 4. Compared to

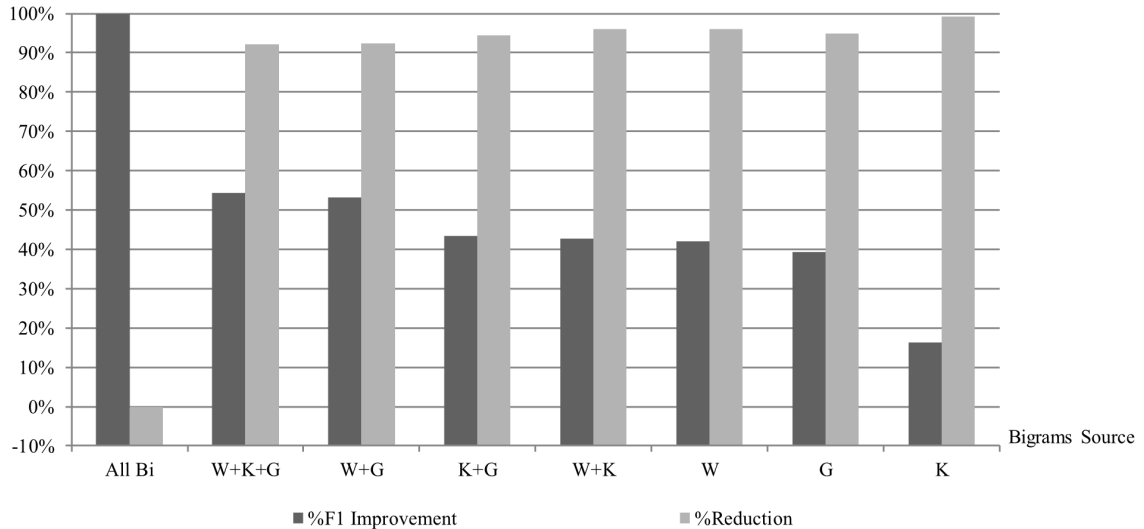


Figure 6: Dimensionality vs. accuracy tradeoff. ”% $F1$ improvement” is the measure of relative gain in accuracy with different bigrams combinations considering all bigrams ”All Bi” are giving $F1$ gain of 100% . % of reduction is the measure of relative gain in dimensionality reduction considering ”All Bi” are giving reduction gain of 0%. (*Wikipedia* ”W”, *Wiktionary* ”K”, *GoogleBooks* ”G”)

the unigrams $F1$ value as a baseline; bigrams, which represent as much as ~ 18 -fold unigrams size, achieved the best improvement (3.18% - 2nd row) which is very close to 3.5% improvement achieved by D’hondt et al. [43].

On the other hand, bigrams from different knowledge sources improved the $F1$ score (with different degrees) when added to unigrams indicating their relevancy. Among the three sources, *Wikipedia* bigrams which is less in size than the unigrams achieved the best $F1$ improvement (1.34% - 3rd row), while *Wiktionary* bigrams which is less than one-sixth of unigrams size achieved the least improvement (0.52% - 4th row).

In addition, looking at the $F1$ improvement for bigrams combinations from different knowledge sources, combining bigrams from *Wikipedia* and *GoogleBooks* was better than all other two-source combinations (1.69% - 7th row). Best improvement

was achieved using bigrams combined from all three sources (1.73% - 9th row). These results indicate that classification accuracy could be controlled by varying source and size of the bigrams features. As shown in Figure 6, the more aggressively dimensionality is reduced, the less improvement in $F1$ could be achieved. In all cases, our approach improved accuracy with different degrees indicating its relevancy and effectiveness.

2.6 Conclusion

In this chapter we introduced a novel approach for dimensionality reduction using free-access, readily available knowledge sources such as *Wikipedia*, *Wiktionary*, and *GoogleBooks*. Unlike the bag of n-grams document representation which introduces very high dimensional feature vectors when adding high order n-grams (e.g, bigrams and trigrams), our approach achieved massive reduction in the number of bigrams features (92%-99%) and thus reducing the representation space and computational complexity enormously, especially with big datasets.

Three interesting properties distinguish our approach from other commonly used feature selection techniques: 1) unlike transformation methods like PCA and LDA, our technique computational complexity is *linearly* proportional to the size of the full feature set, making it scalable to huge and sparse feature spaces, 2) unlike statistical and information theoretic methods like information gain, mutual information, and chi-square our technique is *unsupervised* hence doesn't require class labels to determine feature relevancy, and 3) the set of selected features are obtained from the original feature space and hence easily *interpretable* by humans.

Through experiments, we also introduced a comparative study on the impact of our approach on classification accuracy when using the whole set of bigrams vs. bigrams promoted by the knowledge sources. The results in Figure 6 indicate that there is a *tradeoff* between representation efficiency (dimensionality reduction) and representation effectiveness (classification performance). Thus, learning with the full set of bigrams gave the best results. Nevertheless, our approach gives concrete evidence that, over the unigrams baseline, significant improvement of accuracy is still achievable (1.73%) with massive dimensionality reduction (92.23%).

The results lead us to conclude that we still face the tradeoff between efficiency and effectiveness. We think this tradeoff can be eliminated by more careful preparation of the knowledge-based features; for example, adding selected verb-object pairs from *Wikipedia* articles text, and extracting skip-bigrams from titles rather than bigrams only. The question of what is the smallest subset of features for n dimensional feature space that gives the best result is still open and requires a brute-force search (inspecting the 2^n-1 possible subsets in the worst case).

CHAPTER 3: ENTITY TYPE RECOGNITION USING AN ENSEMBLE OF DISTRIBUTIONAL SEMANTIC MODELS TO ENHANCE QUERY UNDERSTANDING

In this chapter we focus on the increasing the effectiveness of the representation especially short text (e.g., search queries). Understanding short texts such as tweets and search queries is challenging because these text structures has no or limited context, are noisy, and are often ambiguous. Traditional Bag-of-Words (BoW) representation fails to overcome these limitations. For example, the two queries *object oriented developer* and *java programmer*, though have no common words, are highly related and should have similar conceptual representations. To address these issues, we present an ensemble approach for categorizing search query entities in the recruitment domain. Understanding the types of entities expressed in a search query (*Company, Skill, Job Title, School, etc.*) enables more intelligent information retrieval based upon those entities compared to a traditional keyword-based search. Our approach combines clues from different sources of varying complexity in order to collect real-world knowledge about query entities. We employ distributional semantics representations of query entities through two models: 1) contextual vectors generated from concept-based corpora (*Wikipedia*), and 2) word embeddings generated from millions of job postings using Word2Vec. Additionally, our approach utilizes both entity linguistic properties obtained from *WordNet* and ontological properties extracted from *DBpe-*

dia. We evaluate our approach on a dataset created at CareerBuilder¹⁷; one of the largest job boards in the US. The dataset contains entities extracted from millions of job seekers/recruiters search queries, job postings, and resume documents. After constructing the representations ensemble of search entities, we use supervised machine learning to infer search entity types. Empirical results show that our approach outperforms the state-of-the-art Word2Vec model trained on Wikipedia. Moreover, we achieve micro-averaged $F1$ score of 97% using the proposed representations ensemble.

3.1 Motivation

Entity Recognition is an information extraction task which refers to identifying regions of text corresponding to entities. A related sub-task is the Entity Type Recognition (ETR) which refers to categorizing these entities into a predefined set of types [95]. The focus of the majority of ETR research has been on Named Entity Recognition (NER), which typically limits entity types to *Person*, *Location*, and *Organization* [146, 143, 177, 221]. Most techniques used in ETR rely on a mix of local information about the context of the entity and external knowledge usually gained through learning on training data. ETR in search queries is considered extremely important; a Microsoft’s study reported that 71% of queries submitted to their Bing search engine contain named entities somewhere, while 20–30% consist only of named entities [216]. Recognizing the type of entities in queries enables a search engine to understand the intent of users, which subsequently leads to more accurate results being returned. ETR in search queries is very challenging, however, due to the lack

¹⁷<http://www.careerbuilder.com/>

of textual context surrounding the query. Search queries are usually made of just a few words, which is typically not enough context to independently and accurately recognize the types of the entities within a search query. Our research in this chapter is specifically targeted at the problem of ETR within the job search and recruitment domain. Unfortunately, none of the published ETR datasets fully resemble the entity categories within the job search and recruitment domain. Some of the specific entity categories within this domain include *Company*, *Job Title*, *School*, and *Skill*, which all aren't found explicitly within existing ETR datasets. As a result, we can't leverage any existing gazetteers for these entity types.

We introduce a novel system for ETR in search queries which has been applied successfully within the job search and recruitment domain. The proposed system utilizes features collected from *Wikipedia*, *DBpedia*¹⁸, *WordNet* [49], and a corpus of more than 60 million job postings provided by CareerBuilder.

We evaluated this system using a dataset provided by CareerBuilder which contains more than 177K labeled entities. The results demonstrate that our system achieves a 97% micro-averaged *F1* score over all the categories. Because of its high accuracy, CareerBuilder integrated this system into its semantic search engine [6, 7, 100], which improved the quality of search results for tens of millions of job seekers every month.

The system is used within the search engine in two ways: 1) offline, to classify a list of pre-recognized entities extracted from popular queries found in CareerBuilder's search logs, and 2) online, to dynamically classify the search entities within new, previously unseen queries as part of CareerBuilder's semantic query parser.

¹⁸<http://wiki.dbpedia.org/>

The main contributions of our approach are:

1. We introduce a novel approach for generating distributional semantics vectors of named entities in search queries using *Wikipedia* as an intermediate corpus.
2. Our approach is simple and efficient. It outperforms state-of-the-art techniques for distributed representations such as Word2Vec.
3. We evaluate our method on the largest labeled entity type dataset within the recruitment domain achieving a 97% micro-averaged *F1* score.
4. We demonstrate increase in overall system accuracy through an ensemble of features leveraging distributional semantics representations, entity ontologies, and entity linguistic properties.

3.2 Related Work

Both ETR and NER have experienced a surge in the research community in recent years [197, 163, 28, 145, 165, 142, 39]. David et al. [146] and Mansouri et al. [135] presented comprehensive reviews about different approaches for NER including several representations that leverage dictionaries, corpora, and various classification methods.

Guo et al. [65] presented a formulation for both NER and ETR in search queries using a probabilistic approach and latent dirichlet allocation. They represented query terms as words in documents and modeled the entity type classes as topics. They proposed using a weakly supervised learning algorithm to learn the topics, while impressive, their approach was limited to recognizing only one entity per query. Our approach, instead, can accurately identify multiple entities per search query and

recognize their types.

Other approaches which utilize knowledge bases to link named entities in text with corresponding entities in the knowledge bases were presented in [69, 70, 95, 102, 111]. *Wikipedia* has been used extensively as a knowledge base for ETR. Many researchers have utilized *Wikipedia*-based features such as wikilinks, article titles and categories, and graph representations of the inner links between *Wikipedia* pages.

Kazama and Torisawa [95] proposed a methodology which relies on having a *Wikipedia* page whose title is similar to the given entity. After looking up that page, if any, they extracted the category of that entity from the first line in that page. In our case, we couldn't find a *Wikipedia* page for most of the popular queries we have, for example, *java developer* has no corresponding page in *Wikipedia*. Our methodology can handle such cases by looking in *Wikipedia* content not titles for the occurrences of that entity and using the context as a representation in order to recognize the entity type.

Richman and Schone proposed a novel system for multilingual NER [166] . They utilized wikilinks to identify words and phrases that might be entities within text. Once they recognize the entities, they use category links or interlinks to map those entities with English phrases or categories.

Using *Wikipedia* concepts as a representation space for query's intent was introduced by Hu et al. [83]. In this paper each intent domain is represented as a set of *Wikipedia* articles and categories, then each query intent is predicted by mapping the query into the *Wikipedia* representation space.

The system introduced by Nothman et al. [147] transforms links to *Wikipedia* articles into named entity annotations by classifying the target articles into the classic

named entity types *Person*, *Location*, and *Organization*.

Utilizing *Wikipedia* infobox for ETR was presented by Mohamed and Oussalah [144]. The proposed model classifies entities by matching entity attributes extracted from the relevant article infobox with core entity attributes built from *Wikipedia* infobox templates.

The system introduced by Gattani et al. [59] converted *Wikipedia* into a structured knowledge base (KB). In this work, the authors converted *Wikipedia* graph structure into a taxonomy. This was done by finding a single main lineage, called the primary lineage, for each concept. This KB is used later to extract, link, and classify entities mentioned in a Twitter stream.

We consider Laclavík et al. [103] as the most related work to ours. In this work, the authors proposed a system that utilizes *Wikipedia* as an intermediate corpus to categorize search queries. The system works through two phases; in the first phase, a query is mapped to its relevant *Wikipedia* pages by searching an index of *Wikipedia* articles. In the second phase, concepts representing retrieved *Wikipedia* pages are mapped into categories. Though we also utilize a *Wikipedia* search index to retrieve articles related to query entities, our approach utilizes totally different features and entity representation to infer the entity type.

3.3 Methodology

In this section we detail our methodology for recognizing search query entity types. Our approach employs two distributional semantics representations of search entities. Moreover, we use ontological as well as linguistic properties of search entities

to improve the overall system performance. The ultimate goal of our system is to categorize a given search entity into one of four concept classes: *Company*, *Job Title*, *Skill*, and *School*.

3.3.1 System Overview

Prior to performing ETR, it is necessary that we first perform entity recognition on incoming search queries so that we know the entities for which we are trying to identify an entity type. The methodology for recognizing known entities and performing entity extraction from queries is described in AlJadda et al. [5]. First, data mining on historical search query logs combined with collaborative filtering are performed to determine which queries are used commonly together across many users. Then a semantic knowledge base containing the entities and related entities found from within the mined search logs is built and used for entity recognition.

For entities not found in the semantic knowledge base, a language model is created of unigrams, bigrams, and trigrams across a corpus of millions of job posting documents. Leveraging Bayes algorithm, it is possible to dynamically calculate probabilities as to whether any combination of keywords entered into a search query constitute a single phrase or multiple phrases. Based upon the combination of the semantic knowledge base, the Bayes-based phrase identifier, and the query parser, it is possible to successfully identify the correct query parsing including the constituent named entities with accuracy greater than 92%.

After recognizing candidate entities in the user's query, the next stage needed to truly interpret the user's intent correctly is categorizing each of these entities to our

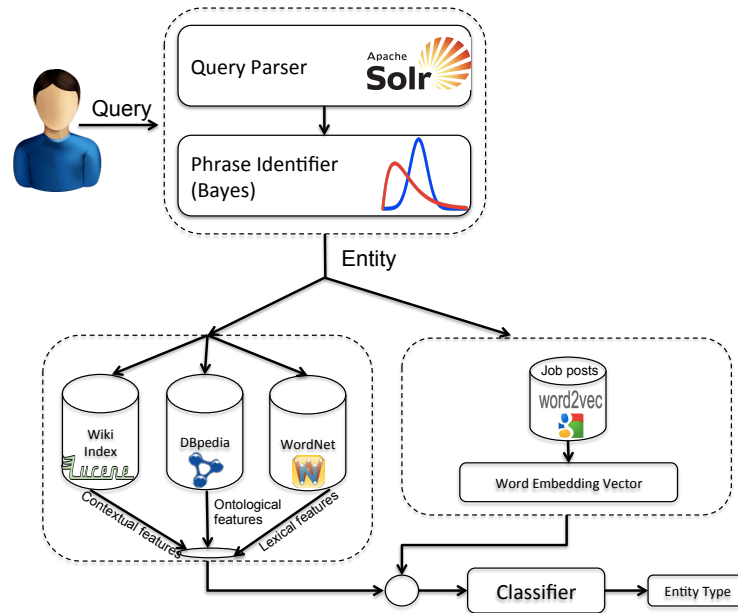


Figure 7: User’s query is passed to the query parser and phrase identifier, which perform entity resolution leveraging a semantic-knowledge-based and a language-model-based probabilistic parsing. The entities are then enriched using an ensemble of representation models based upon external knowledge base (*Wikipedia*), a domain-specific corpus (job postings), ontological features (*DBpedia*), and lexical features (*WordNet*).

predefined types. If a user searches for *“google software engineer java”*, it is critical to understand that the user is looking for a job at *“Google” (Company)* as a *“software engineer” (Job Title)* programming in *“Java” (Skill)*. Without this knowledge of entity types, we will not be able to fully represent the information need of our users within the search system. The following sections will describe our methodology for performing ETR on our identified entities.

3.3.2 The Entity Type Recognition Process

The proposed system combines features from different sources in order to make accurate entity type predictions for a given search entity. This ensemble of features represents our domain-specific knowledge as well as real-world knowledge about the

search entity. We call these features *clues*. Figure 7 shows the system design for how a user’s query is parsed, as well as how the system employs these clues to accurately perform ETR.

The first clue models *real-world contextual information* about the query entity by searching for that entity inside *Wikipedia* using a customized search index. The second clue models *domain-specific knowledge* by building synonyms vector for search entities using the Word2Vec model [138] trained on millions of job postings from CareerBuilder.

Two other clues, using *DBpedia* and *WordNet*, are collected to increase the accuracy and coverage over the *Company* and *Job Title* categories specifically. After collecting all the clues for every known query entity, we combine these features and use them to train an entity classifier on labeled entity samples. The classifier can then be used to categorize new search entities, thus improving our understanding of the query intent for future searches.

3.3.3 Constructing Contextual Vectors

The purpose of this phase is to enrich the contextless search entities with contextual information. In order to do so we map each entity into a distributional semantics vector representation. The vector dimensions represent entity contexts in an intermediate corpus. We use *Wikipedia* as the source for these contextual vectors for all of the search entities which are represented.

As query entities need to be categorized in an online fashion, context vectors are required to be constructed as efficiently as possible. Therefore, we build an inverted

Table 6: Example search entities (left) along with their context vectors (right).

Company	
CareerBuilder	<...market, operate, website, acquired, employment, companies, establishments, ceo...>
Job Title	
Nurse Assistant	<...journalist, worker, secretary, members, politicians, living, people, youth, office...>
Skill	
Adobe Photoshop	<...editor, graphics, developed, image, file, software, application, version, program...>
School	
UNC Charlotte	<...university, north, carolina, college, student, organization, professor, school...>

index of all *Wikipedia* articles as a preprocessing step. We build the index using Apache Lucene¹⁹, an open-source indexing and search engine. For each article we index the *title*, *content*, *length*, and *categories*. We exclude all *disambiguation*, *list of*, and *redirect* pages.

As shown in Equation 3, given an entity e_j we construct its context vector X_{e_j} by first searching for that entity in the search index. Then, from the top n search hits, we retrieve all content words W_i that occur in the same context of e_j within a specific window size in each search hit i . We also retrieve category words C_i of search hits and add them to X_{e_j} .

$$X_{e_j} = \langle w_1, w_2, \dots, c_1, c_2, \dots \rangle : w \in W_i, c \in C_i, i = [1..n] \quad (3)$$

These context vectors represent available real-world knowledge about the given entity. Table 6 shows example search entities along with their context vectors. We can notice that contextual words are very representative for the type of the given entity. Moreover, words from search hits categories augment context words and thus enrich the contextual representation of each entity.

¹⁹<https://lucene.apache.org/>

Table 7: Example search entities (left) along with their synonyms vectors (right).

Company	
CareerBuilder	<...us, software, recruiter, digital...>
Job Title	
Nurse Assistant	<...licensed, registered, nurse, rn, lpn, office, coordinator, lvn, midwife...>
Skill	
Adobe Photoshop	<...dreamweaver, flash, acrobat, macromedia, illustrator, pagemaker...>
School	
UNC Charlotte	<...raleigh, durham, morrisville, hospital, concord, morrisville, durham...>

3.3.4 Constructing Synonyms Vectors

The purpose of this phase is to enrich the search entities with domain-specific knowledge. CareerBuilder has millions of job openings that are posted or modified on daily basis. These postings contain many representative features relevant to the recruitment domain. For example, a typical job posting might contain a job title, job description, required skills, salary information, company information, required experience and education, location...etc.

In order to make use of this information, we use the job postings as an intermediate corpus to train a Word2Vec model [138]. For a given search entity e_j , we generate its synonyms vector S_{e_j} from words that have closest vectors in the trained Word2Vec model based on the cosine similarity.

The distributional semantics vectors generated in this phase represent domain-specific knowledge about a given entity. Table 7 shows the same search entities as in Table 6 along with their corresponding synonyms vectors. We can notice that the *Company* and *School* entity vectors are somewhat poor and unrepresentative. This is because many job postings are missing company information or sometimes company name is only provided without any context. The same problem arises for school information. On the other hand, the synonyms vectors of *Job Title* and *Skill* entities

are very rich and representative. This observation motivated us to combine features for search entities from both contextual and synonyms vectors in a combined vector space.

3.3.5 Entity Ontological Features

Another representative feature is extracted from *DBpedia* by linking search hits (representing *Wikipedia* concepts) to their corresponding entries in the *DBpedia* ontology. We use the *type* property to determine whether the retrieved concept type is one of our targeted categories, specifically *Company*.

After searching for a given entity e_j in the *Wikipedia* index, we retrieve the top n search hits (concepts). Then, we check whether the title of any of these concepts is the same as e_j . If any, we check whether the type of this concept in *DBpedia* ontology is *Company* and subsequently add a new *binary feature* indicating that finding.

Given that companies are already found explicitly in *DBpedia*, why don't we just use the *DBpedia type* feature exclusively for categorizing into the *Company* entity type? There are five reasons we instead choose to combine multiple feature types:

1. *DBpedia* ontology suffers from *low coverage* where many companies in *Wikipedia* don't have a type of *Company* in *DBpedia* (e.g., Boonton Iron Works²⁰, SalesforceIQ²¹).
2. *DBpedia* provides categories for the canonical form of the company name only.

If an entity is searched for using a surface form, the *DBpedia* lookup will fail.

In contrast, *Wikipedia* will generally contain surface forms in the same context

²⁰https://en.wikipedia.org/wiki/Boonton_Iron_Works

²¹<https://en.wikipedia.org/wiki/SalesforceIQ>

as the canonical form (e.g., International Turnkey Systems Group vs. ITS Group²²)

3. As *DBpedia* covers only Wikipedia concepts, it fails to catch companies that do not have a Wikipedia page. Alternatively, these companies will be correctly categorized using their contextual vectors if mentioned in a representative context within *Wikipedia* (e.g., Nutonian).
4. Some companies have a type of *Organization* instead of *Company* in *DBpedia*. Unfortunately, entities belonging to one of our other entity types (*School*) can also be categorized as *Organization* in *DBpedia* (e.g., Athens College). This means that we cannot reliably categorize concepts with the type of *Organization* as *Company*.
5. Finally, there is a *time lag* between *DBpedia* and *Wikipedia*. So, *DBpedia* does not contain the most recent snapshot of *Wikipedia* concepts in its ontology.

3.3.6 Entity Linguistic Features

We utilize the *lexical properties* of search entities to determine whether they belong to one of the target categories, specifically *Job Title*. The motivation behind this approach is the fact that almost all *Job Title* entities contain an *agent noun* (e.g., director, developer, nurse, manager...etc). To determine whether an entity might represent a *Job Title*, we search its words inside the *WordNet* dictionary where all agent nouns are under the <noun.person> lexical file. Upon finding any, we add a new *binary feature* indicating that finding.

²²https://en.wikipedia.org/wiki/International_Turnkey_Systems_Group

While it might be tempting to rely exclusively on the agent noun feature from the *WordNet* lexicon for categorizing *Job Title* entities, two challenges prevent this:

1. Depending solely on the *WordNet* lexicon for categorizing *Job Title* entities would pose limitations on the ETR system for non-English job boards.
2. Not all *Job Title* entities have an agent noun (e.g., staff, faculty).

3.3.7 Building the Prediction Model

To build the ETR model, we use supervised machine learning on a very large labeled set of search entities obtained from CareerBuilder’s search logs. For each discovered search entity e_j , we generate:

1. A contextual vector (X_{e_j}) using the *Wikipedia* index.
2. A synonyms vector (S_{e_j}) using the Word2Vec model.
3. An ontological type (ont_{e_j}) if the entity refers to a *DBpedia* concept. This is a binary feature which is true if *DBpedia* type is company.
4. A lexical type (lex_{e_j}). This is a binary feature which is true if one of the entity terms has a $\langle noun.person \rangle$ type in *WordNet*, i.e., it is an agent noun.

To combine all those features, we follow a simple yet effective approach. First we use the vector space model to generate an entity-word matrix using the both X_{e_j} and S_{e_j} . The generated vectors represent *semantically-related* words to the identified query entities, so it is straightforward to then map each entity as a *document of*

Table 8: Distribution of entities over categories in our dataset

Category	Number of instances
Company	42,934
Job Title	3,608
School	106,153
Skill	25,093

words contained in the entity’s contextual and synonyms vectors. Rows in the entity-word matrix represent entities and columns represent corresponding related words. Secondly, we transform this matrix using term frequency-inverse document frequency (*tf-idf*) weights. Thirdly, we append ont_{e_j} and lex_{e_j} as two additional binary columns to the *tf-idf* entity-word matrix. Finally, we train an entity type classifier on the produced matrix to generate the ETR model.

3.4 Experiments and Results

In this section we present our empirical results. We start by describing the dataset used in experiments and then detail different models developed for ETR along with their results.

3.4.1 Dataset

We built our ETR models using the largest labeled entity dataset owned by CareerBuilder. The dataset contains more than 177K labeled entities distributed over four categories as shown in Table 8. These entities were obtained from CareerBuilder’s search logs, job postings, and resumes, and were manually reviewed by annotators working at CareerBuilder.

3.4.2 Experimental Setup

We conducted several experiments in order to evaluate the performance of the ETR system with different models. We started by evaluating models built from a single feature source, i.e., contextual vectors or entity synonyms vectors. Then we evaluated a model built using an ensemble of both of these vectors. Finally, we evaluated a model which combines both vectors plus the entity’s ontological and lexical features (i.e., ont_{e_j} and lex_{e_j} respectively).

To assess the effectiveness of our approach, we built two baseline models. The first one is the Bag-of-Words (BoW) model which depends solely on words that appear in search entities as features without any contextual enrichment. The second model ($wiki_w$) is a distributional semantics model built by training Word2Vec on *Wikipedia*. After Word2Vec produces the word vectors, word synonyms vectors of search entities are generated as described in Section 3.3.4. We then generate a *tf-idf* entity-word matrix from these vectors as described in Section 3.3.7.

We built the *Wikipedia* search index using the English *Wikipedia* dump of March 2015²³. The total uncompressed XML dump size was about 52GB representing about 7 million articles. We extracted the articles using a modified version of the *Wikipedia* Extractor²⁴. Our version²⁵ extracts articles as plain text, discarding references to images and tables. We discarded the *References* and *External Links* sections (if any). We pruned all articles which are not under the main namespace, and excluded all *disambiguation*, *list of*, and *redirect* pages as well. Eventually, our index contained

²³<https://dumps.wikimedia.org/enwiki/20150304/>

²⁴http://medialab.di.unipi.it/wiki/Wikipedia_Extractor

²⁵<https://github.com/walid-shalaby/wikiextractor>

about 4 million documents.

While searching the *Wikipedia* index, we search both content and title fields. For efficiency, we limit retrieved results to the top 3 hits which have a minimum length of 100 bytes.

To build the word embedding vectors, we trained Word2Vec on more than 60 million job postings from CareerBuilder. We used Apache Spark’s scalable machine learning library (MLlib²⁶) which has an implementation of Word2Vec in Scala. We configured the parameters of the Word2Vec model as follows: minimum word count = 50, number of iterations (epoch)=1, vector size = 300, and number of partitions = 5000. The model took about 32 hours to fit on one of CareerBuilder’s Hadoop clusters with 69 data nodes, each having a 2.6 GHz AMD Opteron Processor with 12 to 32 cores and 32GB to 128GB RAM.

Finally, we evaluate all the ETR models using a Support Vector Machine (SVM) classifier with linear kernel, leveraging the scikit-learn library [152]. Because entity instance frequencies over categories is a bit skewed and to avoid overfitting, we configured the classifier to use a different regularization value for each category relative to category frequencies. For each model we report Precision (P), Recall (R), and their harmonic mean ($F1$) scores. Results are calculated using 10-fold cross-validation over the labeled entities dataset. And folds were randomly generated using stratified sampling.

²⁶<https://spark.apache.org/mllib/>

Table 9: Performance of the contextual vectors ETR model ($wiki_x$) on labeled entities compared to baseline models using 10-fold cross-validation. BoW is the Bag-of-Words model, and $wiki_w$ is Word2Vec trained on Wikipedia.

Category	Company			Job Title			School			Skill		
	<i>P</i>	<i>R</i>	<i>F1</i>	<i>P</i>	<i>R</i>	<i>F1</i>	<i>P</i>	<i>R</i>	<i>F1</i>	<i>P</i>	<i>R</i>	<i>F1</i>
BoW	91.46	79.72	85.19	84.92	90.08	87.42	99.07	94.23	96.59	66.07	91.04	76.57
$wiki_w$	88.92	92.23	90.54	85.85	93.82	89.66	98.92	96.42	97.66	87.36	88.15	87.75
$wiki_x$	95.41	96.55	95.98	86.27	88.30	87.28	98.93	98.11	98.52	91.99	92.42	92.21

3.4.3 Results

Table 9 shows the results obtained from the baseline models compared to the contextual vectors model using 10-fold cross-validation on the labeled entities dataset.

The first baseline model is the BoW . This model gives relatively lower $F1$ scores on all categories as shown in Table 9. Due to the absence of contextual information, this model fails to generalize well with unseen entities, as they contain terms that are not in the model’s feature space. This is very clear with categories that have high naming variations (i.e., $Company$ and $Skill$). BoW performs relatively well on $Job Title$ as it has limited naming variations. It also performs very well on $School$ entities as they have common naming conventions (e.g., university, school, institute...etc).

The second baseline model is $wiki_w$ which is built by training Word2Vec on *Wikipedia*. This model utilizes contextual features inferred from word distributions, hence it performs better than BoW on all categories. As shown in Table 9, the $wiki_w$ $F1$ score is higher than BoW by more than 5% on $Company$, 2% on $Job Title$, 1% on $School$, and 11% on $Skill$. Those results indicate the viability of distributional semantics representations for ETR of search entities.

The third model is $wiki_x$ which is built using contextual vectors generated by searching the *Wikipedia* index. It retrieves search entity contexts and category in-

formation from search hits and then utilizes them as learning features. As shown in Table 9, this novel approach outperforms both *BoW* and *wiki_w* models substantially on *Company* and *Skill*. It also performs slightly better on *School*. These results indicate the effectiveness of the *wiki_x* model in recognizing these categories accurately.

It is important to mention that, though both the *wiki_x* and *wiki_w* models use *Wikipedia* as an intermediate corpus to learn word representations, the *wiki_x* representations are more successful for the ETR task. Compared with the *wiki_w* model, the *F1* scores of the *wiki_x* model increased on the *Company* class by 5%, on the *School* class by 1%, and on the *Skill* class by 5%.

The *Job Title* category is the only example where the *wiki_w* model performed better (by 2%) than the *wiki_x* model. A closer look at the scores reveals that, the *wiki_x* model is more accurate than the *wiki_w* model as it has a higher *P* score. The *wiki_w* model, however, has better coverage as it has a higher *R* score. Considering the small size of the *Job Title* category ($\sim 3,600$ entities), that difference in recall cannot be considered substantial.

The results in Table 9 prove empirically that, for ETR of search entities, our novel approach for modeling real-world knowledge using contextual representations outperforms Word2Vec, the state-of-the-art for distributional semantics representations, even though both use the same intermediate corpus (*Wikipedia*). Moreover, our method is much simpler and more efficient than Word2Vec as it doesn't require optimizing an objective function for learning word embeddings.

In order to increase the overall system performance, we built four ETR models that combine features from different sources as described in Section 3.3.7. We first built

Table 10: Performance of different ETR models on the labeled entity dataset using ensemble of features.

Category	Company			Job Title			School			Skill		
Metric	<i>P</i>	<i>R</i>	<i>F1</i>	<i>P</i>	<i>R</i>	<i>F1</i>	<i>P</i>	<i>R</i>	<i>F1</i>	<i>P</i>	<i>R</i>	<i>F1</i>
<i>wiki_x</i>	95.41	96.55	95.98	86.27	88.30	87.28	98.93	98.11	98.52	91.99	92.42	92.21
<i>wiki_x, job_w</i>	95.64	96.73	96.18	88.32	91.99	90.12	99.16	98.20	98.68	92.45	93.17	92.81
<i>wiki_x, job_w, ont</i>	96.38	96.72	96.55	87.94	92.13	89.98	99.14	98.25	98.69	92.33	93.95	93.13
<i>wiki_x, job_w, lex</i>	95.67	96.68	96.17	88.34	92.82	90.53	99.16	98.21	98.68	92.49	93.14	92.81
<i>wiki_x, job_w, lex, ont</i>	96.41	96.72	96.56	88.35	92.91	90.57	99.15	98.23	98.69	92.31	93.99	93.14

job_w which models domain-specific knowledge of search entities. The *job_w* model is built by training Word2Vec on the textual content of millions of job postings.

As shown in Table 10, we combined both the contextual vectors (*wiki_x*) and the synonyms vectors (*job_w*) and built an ensemble of the two models (*wiki_x, job_w*). The ensemble improved the results over *wiki_x* across all categories. the largest improvement was on *Job Title*, which saw a 3% improvement in *F1* score. More importantly, this ensemble outperforms the *wiki_w* and *BoW* models on all categories.

To further increase system accuracy on *Company* class, we incorporated the *DBpedia* ontological type of search entity (*ont*) with the contextual and synonyms vectors as described in Section 3.3.5. This ensemble (*wiki_x, job_w, ont*), as shown in Table 10, increased *F1* score on *Company* by $\sim 0.4\%$.

The third ensemble is (*wiki_x, job_w, lex*). It aims at increasing system accuracy on *Job Title* class by incorporating entity’s lexical features (*lex*) as described in Section 3.3.6. As shown in Table 10, the *F1* score on *Job Title* increased by $\sim 0.6\%$ when incorporating this feature.

Finally, we combined all features generating an ensemble of contextual vectors, synonyms vectors, ontological features, and linguistics features (*wiki_x, job_w, lex, ont*). As shown in Table 10, this model produced the best *F1* scores on all categories among all the aforementioned models.

3.5 Conclusion

In this chapter we focused on improving the effectiveness of the representation of short texts, specifically search queries. We presented an effective approach for entity type recognition (ETR) of search query entities in the job search and recruitment domain using concept knowledge bases along with concepts lexical and ontological properties.

The proposed novel ensemble of features enriches short query entities representation with real-world and domain-specific knowledge. The ensemble entity representation contains features representing: 1) contextual information in *Wikipedia*, 2) embedding information in millions of job postings, 3) class type in *DBpedia* for *Company* entities, and 4) linguistic properties in *WordNet* for *Job Title* entities. This approach coincides directly with our overall aim employing concept KBs, concept properties, and distributed representations in order to enhance the representation and subsequently performance of text analysis systems.

Our approach is novel and distinct from other ETR approaches. To our knowledge, generating distributional semantics vectors of query entities using contextual information from *Wikipedia* as a search index was not reported before in the literature.

Evaluation results on a dataset of more than 177K search entities were very promising. The results showed that our *Wikipedia*-based model outperforms the state-of-the-art Word2Vec model trained on *Wikipedia* on three out of four target entity categories. Moreover, our ensemble representation could achieve 97% micro-averaged *F1* score on the four entity types outperforming the Word2Vec baseline by 6% on

Company, 1% on Job Title, 1% on School, and 5% on Skill.

In terms of performance, our system takes 30ms per entity type request, making it efficient and appropriate for serving online search queries.

Our system has been integrated within CareerBuilder's semantic search engine, which improved the quality of search results for tens of millions of job seekers every month.

CHAPTER 4: MINED SEMANTIC ANALYSIS

In this chapter we focus on increasing the effectiveness of concept-based semantic representation models (aka explicit concept space or bag-of-concepts models). These models have proven efficacy for text representation in many natural language and text mining applications. The idea is to embed textual structures into a semantic space of concepts which captures the main ideas, objects, and the characteristics of these structures. We start by surveying existing techniques for bag-of-concepts representations and their applications along with some of their limitations. Then we introduce Mined Semantic Analysis (MSA); our novel bag-of-concepts model which aim to enhance the expressiveness of the concept-based representations through utilizing unsupervised data mining techniques in order to discover concept-concept associations. These associations are then used to enrich the bag-of-concepts with more related concepts.

We demonstrate the efficacy of MSA on two tasks: 1) measuring lexical semantic relatedness, and 2) short text similarity. Empirical results show superior performance of MSA over other bag-of-concepts and distributed representations on the two tasks.

4.1 Text Conceptualization

Vector-based semantic representation models are used to represent textual structures (words, phrases, and documents) as *multidimensional vectors*. Typically, these

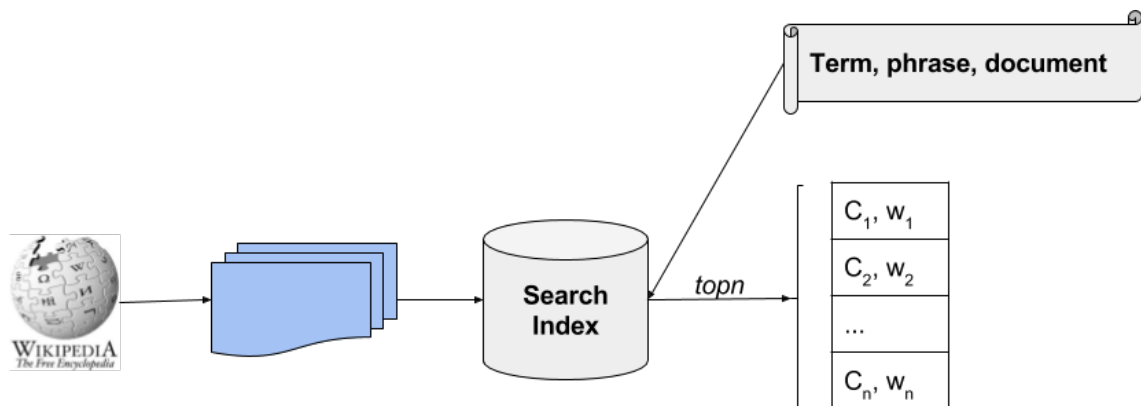


Figure 8: The general architecture of *Wikipedia*-based concept space representation models. The concept space is defined by all *Wikipedia* article titles. The concept vector is created from the top n hits of searching a *Wikipedia* inverted index with the given text.

models utilize textual corpora and/or Knowledge Bases (KBs) in order to extract and model real-world knowledge. Once acquired, any given text structure is represented as a real-valued vector in the semantic space. The goal is thus to accurately place semantically similar structures close to each other in that semantic space, while placing dissimilar structures far apart.

Explicit concept space models are one of these *vector-based representations* which are motivated by the idea that, high level cognitive tasks such learning and reasoning are supported by the knowledge we acquire from *concepts* [184]. Therefore, such models utilize concept vectors (aka Bag-of-Concepts (BoC)) as the underlying semantic representation of a given text through a process called *conceptualization*, which is mapping the text into relevant concepts capturing its main ideas, objects, events, and their characteristics. The concept space typically includes concepts obtained from concept-rich KBs such as *Wikipedia*, *Probase* [212], and others. Once the concept vectors are generated, similarity between two concept vectors can be computed using a

suitable similarity measure such as *cosine*. Figure 8 shows the general architecture of the vector-based concept space models utilizing *Wikipedia* as the source of concepts.

The BoC vector is a multidimensional sparse vector whose dimensionality is the same as the number of concepts in the employed KB (typically millions). Formally, given a text snippet $T = \{t_1, t_2, \dots, t_n\}$ of n terms where $n \geq 1$, and a concept space $C = \{c_1, c_2, \dots, c_N\}$ of size N . The BoC vector $\mathbf{v} = \{w_1, w_2, \dots, w_N\} \in \mathbb{R}^N$ of T is a vector of weights of each concept where each weight w_i of concept c_i is calculated as in equation 4:

$$w_i = \sum_{j=1}^n f(c_i, t_j), 1 \leq i \leq N \quad (4)$$

Here $f(c, t)$ is a *scoring function* which indicates the degree of *association* between term t and concept c . For example, Gabrilovich and Markovitch [56] proposed Explicit Semantic Analysis (ESA) which uses Wikipedia articles as concepts and the *tf-idf* score of the terms in these article as the association score. Another scoring function might be the co-occurrence count or Pearson correlation score between t and c .

Typically, the *cosine* similarity measure is used compute the similarity between a pair of BoC vectors \mathbf{u} and \mathbf{v} . Because the concept vectors are very sparse and for space efficiency, we can rewrite each vector as a vector of tuples (c_i, w_i) . Suppose that $\mathbf{u} = \{(c_{n_1}, u_1), \dots, (c_{n_{|\mathbf{u}|}}, u_{|\mathbf{u}|})\}$ and $\mathbf{v} = \{(c_{m_1}, v_1), \dots, (c_{m_{|\mathbf{v}|}}, v_{|\mathbf{v}|})\}$, where u_i and v_j are the corresponding weights of concepts c_{n_i} and c_{m_j} respectively. And n_i, m_j are the indices of these concepts in the concept space C such that $1 \leq n_i, m_j \leq N$. Then, the

In this paper we describe our work on cognitive assistance (Cog) technology in the *innovation analytics* domain. We propose a framework for innovation analytics and *management* using Mined *Semantic Analysis* (MSA). Our goal is to build a semantic driven visual interactive analytics engine that provides insights on *innovation data* using conceptual knowledge derived from huge *unstructured textual knowledge corpora* (e.g., Wikipedia). Throughout the paper we demonstrate a case study utilizing our framework for providing computational assists on *competitive intelligence* by automatically defining the *innovation portfolio* of an organization, and using that information to identify other key players with similar portfolios which could be candidates for *acquisition*

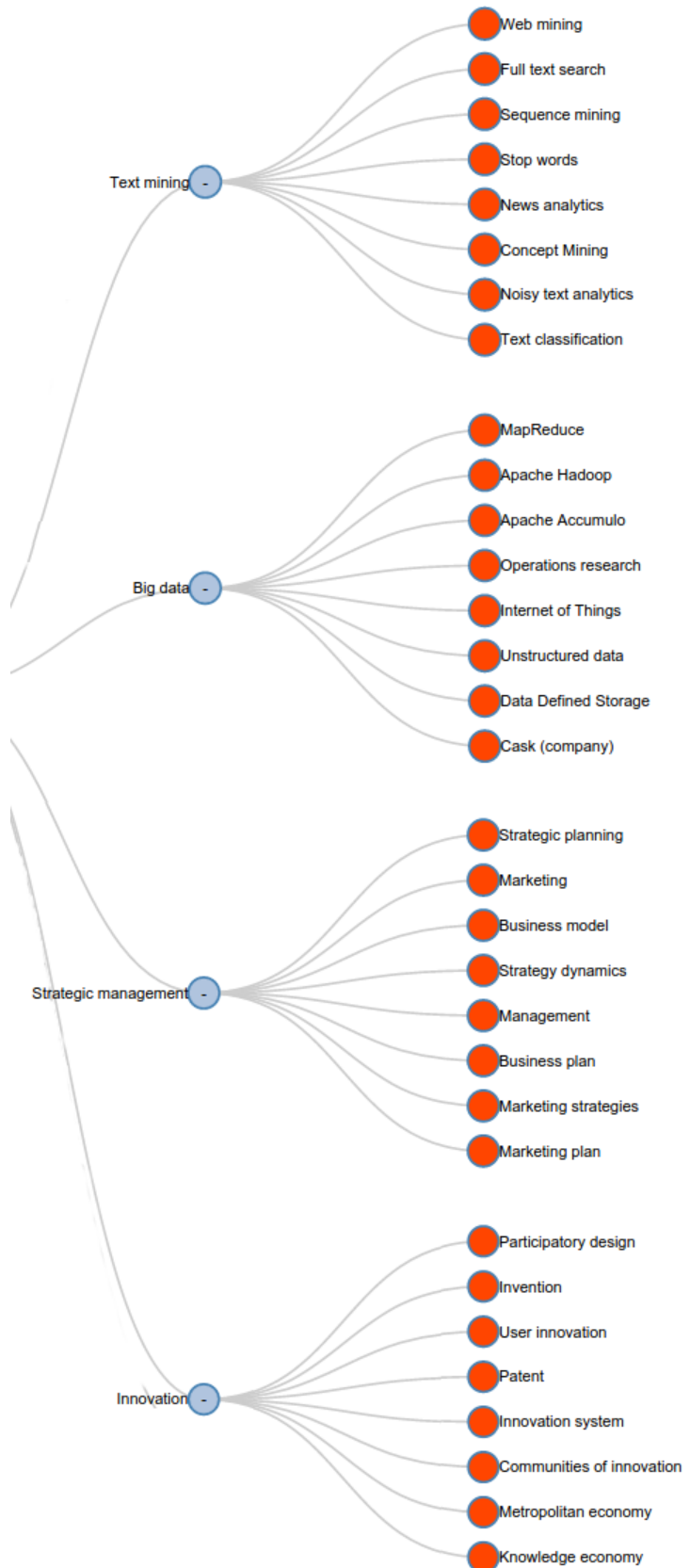


Figure 9: The concept graph representation of the abstract section of Shalaby and Zadrozny [178] (text on the left). Light blue nodes are explicit concepts and red nodes are associated concepts.

similarity score can be written as in equation 5:

$$Sim_{cos}(\mathbf{u}, \mathbf{v}) = \frac{\sum_{i=1}^{|\mathbf{u}|} \sum_{j=1}^{|\mathbf{v}|} \mathbb{1}(n_i=m_j) u_i v_j}{\sqrt{\sum_{i=1}^{|\mathbf{u}|} u_i^2} \sqrt{\sum_{j=1}^{|\mathbf{v}|} v_j^2}} \quad (5)$$

where $\mathbb{1}$ is the indicator function which returns 1 if $n_i=m_j$ and 0 otherwise.

Expressiveness is one of the big advantages of explicit concept representations compared to implicit and distributed representations. As the representation is composed of a vector of concepts, it can be easily *understood* by humans and thus *easier to interact with*. Figure 9 shows the concept vector generated using MSA (described in section 4.2) from the abstract text of Shalaby and Zadrozny [178]. It is important to mention that, the explicit concepts (blue nodes) are ranked top-down according to their relevance to the seed text. The implicit concepts (red nodes) of each explicit concept are also ranked top-down according to their relevance to the explicit concept based on the strength of the association between them.

As we can see, the representation captures four main semantically related concepts of the given text (blue nodes) including: *Text mining*, *Big data*, *Strategic management*, and *Innovation*. As we will show in section 4.2, MSA could enrich these four basic concepts with more related concepts (red nodes). These implicit concepts serve as a powerful mechanism for *concept expansion*. They augment the knowledge required to capture key ideas expressed in the seed text in multiple ways offering hypernymy/abstraction (*Strategic Management* and *Management*), hyponymy/specificity (*Text mining* and *Text classification*), synonymy (*Innovation* and *Invention*), and relatedness/associativity (*Big data* and *Internet of Things*).

4.1.1 Text Conceptualization Methods

Humans understand languages through multi-step cognitive processes which involves building rich models of the world and making multi-level generalizations from the input text [209]. One way of automating such generalizations is through *text conceptualization*. Either by *extracting basic level concepts* from the input text using concept KBs [96, 184], or *mapping* the whole input into a concept space that captures its semantics [56, 77, 23].

As mentioned in Section 1.1.4, text conceptualization comes in two flavors. One line of conceptualization research uses semi-structured KBs such as *Wikipedia* in order to construct the concept space which is defined by all *Wikipedia* article titles. Another research direction uses more structured concept KBs such as *Microsoft Knowledge Graph (Probase)*²⁷ [212] which consists of millions concepts and their relationships (basically an is-a hierarchy).

4.1.2 Vector-based Concept Space Models

A closely related method to our proposed Mined Semantic Analysis (MSA) method is Explicit Semantic Analysis (ESA) [56]. ESA constructs the concept space of a term by *searching an inverted index* of term-concept co-occurrences. ESA is mostly the traditional vector space model applied to *Wikipedia* articles. ESA is effective in retrieving concepts which explicitly mention the target search terms in their content. However, it fails to identify other *implicit concepts* which do not contain the search terms. MSA bridges this gap by *mining for concept-concept associations* and thus

²⁷<https://concept.research.microsoft.com>

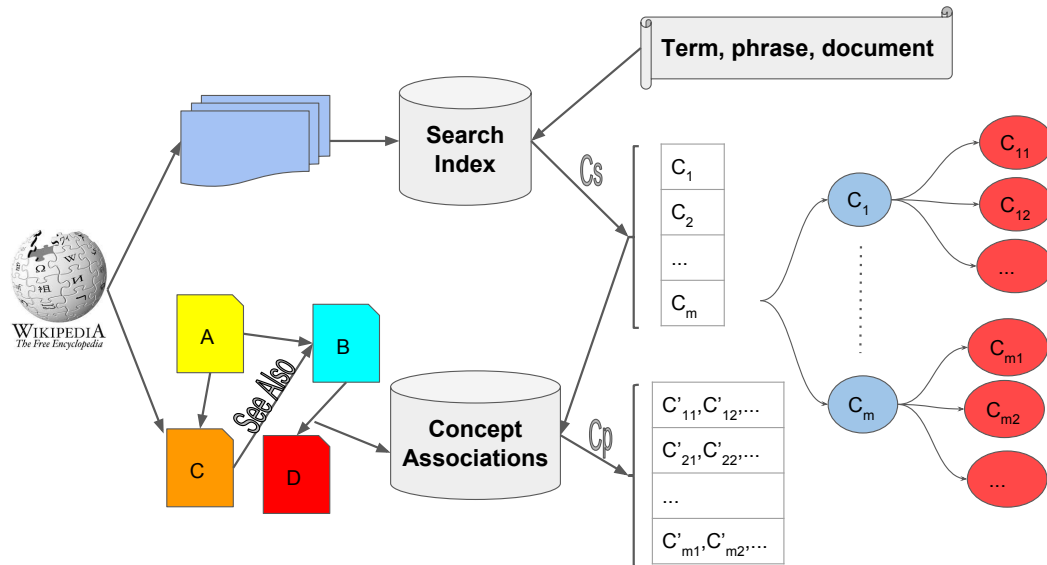


Figure 10: MSA generates the concept vector of a given text structure through: 1) explicit concept retrieval from the inverted index (top); and 2) concept expansion from the concept-concept associations repository (bottom).

augmenting the concept space identified by ESA with more relevant concepts.

Salient Semantic Analysis (SSA) was proposed by Hassan and Mihalcea [77]. It uses *Wikipedia* concepts to build semantic profiles of words. SSA is more conservative than ESA as it defines word meaning by its immediate context and therefore might yield concepts of higher relevancy. However, it is still limited to surface semantic analysis because it, like ESA, utilizes only *direct associations* between words and concepts and fails to capture other implicit concepts not directly co-occurring with corpus words in the same context.

In order to discover indirect relationships between concepts, Radinsky et al. [160] proposed Temporal Semantic Analysis (TSA) which works by extending ESA's concept space to include temporal usage patterns of discovered concepts. These temporal patterns are collected from a corpus of temporally-ordered articles (e.g., newspaper

Table 11: The concept representation of "*Computational Linguistics*"

Explicit Concepts	Implicit Concepts
Parse tree	Universal Networking Language
Temporal annotation	Translation memory
Morphological dictionary	Systemic functional linguistics
Textalytics	Semantic relatedness
Bracketing	Quantitative linguistics
Lemmatization	Natural language processing
Indigenous Tweets	Internet linguistics
Statistical semantics	Grammar induction
Trebank	Dialog systems
Light verb	Computational semiotics

archive). Both MSA and TSA share a common goal; they try to *complement* the concept space with information that *uncovers implicit concept associations*. However, they follow totally different methodologies for achieving that goal. TSA exploits temporal dynamics of concept usage, while MSA exploits mining the semantic space of each concept as expressed in its associations with other concepts.

4.2 Mined Semantic Analysis

We call our approach Mined Semantic Analysis (MSA) as it utilizes *unsupervised data mining* techniques in order to discover the concept space of textual structures. The motivation behind our approach is to mitigate a notable gap in previous concept space models which are limited to direct associations between words and concepts. Therefore those models lack the ability to transfer the association relation to other implicit concepts which contribute to the meaning of these words.

Figure 10 shows MSA’s architecture. In a nutshell, MSA generates the concept vector of a given text by utilizing two repositories created offline: 1) a search index of *Wikipedia* articles, and 2) a concept-concept associations repository created by

mining the "See also" link graph of *Wikipedia*. First, the explicit concept vector is constructed by retrieving concepts (titles of articles) explicitly mentioning the given text. Second, implicit concepts associated with each of the explicit concepts are retrieved from the associations repository and used to augment the concept vector.

To demonstrate our approach, we provide an example of exploring the concept representation of "*Computational Linguistics*" (Table 11). Column 1 shows the explicit concepts retrieved by searching *Wikipedia*²⁸. Column 2 shows the same explicit concepts in column 1 enriched by the implicit concepts from the associations repository. As we can notice, those implicit concepts augment the explicit concept space by more related concepts which contribute to understanding "*Computational Linguistics*". It is worth mentioning that not all implicit concepts are equally relevant, therefore we also propose an automated mechanism for ranking those concepts in a way that reflects their relatedness to the original search term.

4.2.1 The Search Index

MSA starts constructing the concept vector of term(s) by searching for an initial set of candidate explicit concepts. For this purpose, we build a search index of a concept-rich corpus such as *Wikipedia* where each article represents a concept. This is similar to the idea of the inverted index introduced in ESA [56]. We build the index using *Apache Solr*²⁹, an open-source indexing and search engine. For each article we index the *title*, *content*, *length*, number of *outgoing links*, and the "See also" section.

During search we use some parameters to tune the search space. Specifically, we

²⁸We search *Wikipedia* using a term-concept inverted index and limit the search space to articles with min. length of 2k and max. title length of 3 words.

²⁹<http://lucene.apache.org/solr/>

define the following parameters to provide more control over search:

1. **Article Length (L):** Minimum length of the *Wikipedia* article in characters excluding sections like "*References*", "*See also*", "*Categories*", ...etc.
2. **Outdegree (O):** Minimum number of outgoing links per article.
3. **Title Length (τ):** This threshold is used to prune all articles that have long titles. It represents the maximum number of words in the title. E.g, if $\tau=3$, then all articles with more than three words in title will be excluded from the initial candidate concepts.
4. **Number of Concepts (M):** Maximum number of concepts (articles) to retrieve as initial candidate concepts.

4.2.2 Association Rules Mining

In order to discover the implicit concepts, we employ the *Apriori* algorithm for association rule learning [2] to *learn implicit relations* between concepts using *Wikipedia's* "*See also*" link graph.

Formally, given a set of concepts $C = \{c_1, c_2, \dots, c_N\}$ of size N (i.e., all *Wikipedia* articles). We build a dictionary of transactions $T = \{t_1, t_2, t_3, \dots, t_M\}$ of size M such that $M \leq N$. Each transaction $t \in T$ contains a subset of concepts in C . t is constructed from each article in *Wikipedia* that contains at least one entry in its "*See also*" section. For example, if an article representing concept c_1 with entries in its "*See also*" section referring to concepts $\{c_2, c_3, \dots, c_n\}$, a transaction $t = \{c_1, c_2, c_3, \dots, c_n\}$ will be added to T . A set of rules R is then created by mining T . Each rule $r \in R$ is

defined as in equation 6:

$$r(s, f) = \{(X \Rightarrow Y) : X, Y \subseteq C \text{ and } X \cap Y = \emptyset\} \quad (6)$$

Both X and Y are subsets of concepts in C . X are called the antecedents of r and Y are called the consequences. Rule r is parameterized by two parameters: 1) *support* (s) which indicates how many times both X and Y co-occur in T , and 2) *confidence* (f) which is s divided by number of times X appeared in T .

After learning R , we end up having concept(s)-concept(s) associations. Using such rules, we can determine the strength of those associations based on s and f .

As the number of rules grows exponentially with the number of concepts, we define the following parameters to provide more fine grained control on participating rules during explicit concept expansion:

1. **Consequences Size** ($|Y|$): Number of concepts in rule consequences.
2. **Support Count** (σ): It defines the minimum number of times antecedent concept(s) should appear in T .
3. **Minimum Rule Support** (ϵ): It defines the minimum strength of the association between rule concepts. For example, if $\epsilon=2$, then all rules whose support $s \geq 2$ will be considered during concept expansion.
4. **Minimum Confidence** (ν): It defines the minimum strength of the association between rule concepts compared to other rules with same antecedents. For example, if $\nu=0.5$, then all rules whose confidence $f \geq 0.5$ will be considered

during concept expansion. In other words, consequent concept(s) must have appeared in at least 50% of the times antecedent concept(s) appeared in T .

4.2.3 Constructing the Concept Vector

Given a set of concepts C of size N , MSA constructs the bag-of-concepts vector C_t of term(s) t through two phases: *Search* and *Expansion*. In the search phase, t is represented as a search query and is searched for in the *Wikipedia* search index. This returns a weighted set of articles that best matches t based on the vector space model. We call the set of concepts representing those articles C_s and is represented as in equation 7:

$$C_s = \{(c_i, w_i) : c_i \in C \text{ and } i \leq N\} \quad (7)$$

$$\text{subject to : } |title(c_i)| \leq \tau, \text{ length}(c_i) \geq L, O_{c_i} \leq O, |C_s| \leq M$$

Note that we search all articles whose content length, title length, and outdegree meet the thresholds L , τ , O respectively. The weight of c_i is denoted by w_i and represents the match score between t and c_i as returned by the search engine.

In the expansion phase, we first prune all the search concepts whose support count is below the threshold σ . We then use learned association rules to expand each remaining concept c in C_s by looking for its associated set of concepts in R . Formally, the expansion set of concepts C_p is obtained as in equation 8:

$$C_p = \bigcup_{c \in C_s, c' \in C} \{(c', w') : \exists r(s, f) = c \Rightarrow c'\} \quad (8)$$

$$\text{subject to : } |c'| = |Y|, s \geq \epsilon, f \geq v$$

Note that we add all the concepts that are implied by c where this implication meets

the support and confidence thresholds (ϵ, ν) respectively. The weight of c' is denoted by w' . Two weighting mechanisms can be employed here: 1) inheritance; where c' will have the same weight as its antecedent c , and 2) proportional; where c' will have prorated weight $w' = f * w$ based on the confidence f of $c \Rightarrow c'$.

Finally, all the concepts from search and expansion phases are merged to construct the concept vector C_t of term(s) t . We use the disjoint union of C_s and C_p to keep track of all the weights assigned to each concept as in equation 9:

$$C_t = C_s \sqcup C_p \quad (9)$$

4.2.4 Concept Weighting

Any concept $c \in C_t$ should have appeared in C_s at most once, however c might have appeared in C_p multiple times with different weights. Suppose that $\{w_1, \dots, w_n\}$ denotes all the weights c has in C_t where $n < |C_p| + 1$, then we can calculate the final weight w by aggregating all the weights as in equation 10:

$$w = \sum_{i=1}^n w_i \quad (10)$$

Or we can take the maximum weight as in equation 11:

$$w = \max_i w_i \quad 1 \leq i \leq n \quad (11)$$

The rationale behind weight aggregation (equation 10) is to ensure that popular concepts which appear repeatedly in the expansion list will have higher weights than those which are less popular. As this scheme might favor popular concepts even if they appear in the tail of C_s and/or C_p , we propose selecting only the maximum

weight (equation 11) to ensure that top relevant concepts in both C_s and C_p still have the chance to maintain their high weights.

4.2.5 Relatedness Scoring

We apply the cosine similarity measure in order to calculate the relatedness score ($Rel_{cos}(t_1, t_2)$) between a pair of concept vectors \mathbf{u} and \mathbf{v} of terms t_1 and t_2 as in equation 5.

Similar to Hassan and Mihalcea [77], we include a normalization factor λ as the cosine measure gives low scores for highly related terms due to the sparsity of their concept vectors. Other approaches for dealing with vector sparsity will be explored in Chapter 6. Using λ , the final relatedness score will be adjusted as in equation 12:

$$Rel(t_1, t_2) = \begin{cases} 1 & Rel_{cos}(t_1, t_2) \geq \lambda \\ \frac{Rel_{cos}(t_1, t_2)}{\lambda} & Rel_{cos}(t_1, t_2) < \lambda \end{cases} \quad (12)$$

4.3 Experiments on Semantic Similarity and Relatedness

Measuring the semantic similarity/relatedness between text structures (words, phrases, and documents) has been the standard evaluation task for almost all proposed semantic representation models. Although semantic similarity and relatedness are often used interchangeably in the literature, they do not represent the same task [22]. Evaluating genuine similarity is, and should be, concerned with measuring the similarity or resemblance in meanings and hence focuses on the synonymy relations (e.g., *smart, intelligent*). Relatedness, on the other hand, is more general and covers broader scope as it focuses on other relations such as antonymy (*old, new*), hypernymy

(*red,color*), and other functional associations (*money,bank*).

Semantic relatedness has many applications in natural language processing and information retrieval for addressing problems such as word sense disambiguation, paraphrasing, text categorization, semantic search, and others.

Semantic relatedness methods often develop a mapping model which represents each linguistic term as a vector derived from its contextual information in a large corpus of text or knowledge base [12, 199, 77, 56, 105]. After constructing such semantic vectors, relatedness is calculated using an appropriate vector similarity measure (e.g., cosine similarity).

Before applying MSA to the task of information retrieval of technical text, we evaluate the efficacy of MSA concept vectors on two text analysis tasks: 1) measuring lexical semantic relatedness between pairs of words, and 2) evaluating short text similarity between pairs of short text snippets. These tasks test the agreement of the induced representation with the human judgments on commonsense concepts.

4.3.1 Lexical Semantic Relatedness

4.3.1.1 Datasets

We evaluate MSA’s performance on benchmark datasets for measuring lexical semantic relatedness. Each dataset is a collection of word pairs along with human judged similarity/relatedness score for each pair.

RG³⁰: A similarity dataset created by Rubenstein and Goodenough [171]. It contains 65 noun pairs. Similarity judgments of each pair were conducted by 51 subjects.

³⁰<https://github.com/mfaruqui/eval-word-vectors/tree/master/data/word-sim>

Judgments range from 0 (very unrelated) to 4 (very related). Pilehvar and Navigli [155] reported the highest performance on this dataset by creating a semantic network from *Wiktionary*.

MC³⁰: A similarity dataset created by Miller and Charles [141]. It contains 30 noun pairs taken from *RG* dataset. Similarity judgments were done by 38 subjects at the same scale as *RG*. Camacho-Collados et al. [23] reports the highest performance on *MC* by integrating knowledge from *Wikipedia* and *Wordnet*.

WS³⁰: A relatedness dataset of 353 word pairs created by Finkelstein et al. [50]. Relatedness score for each pair ranges from 0 (totally unrelated) to 10 (very related). Annotators were not instructed to differentiate between similarity and relatedness. Halawi et al. [67] reports the highest performance on *WS* using a supervised model combined with constraints of known related words.

WSS & WSR³⁰: Agirre et al. [1] manually split *WS* dataset into two subsets to separate between similar words (*WSS* of 203 pairs), and related words (*WSR* of 252 pairs). Baroni et al. [12] reports the highest performance on both datasets using the popular neural embeddings model *Word2Vec* [138].

MEN³¹: A relatedness dataset created by Bruni et al. [21]. We use the test subset of this dataset which contains 1000 pairs. Relatedness scores range from 0 (totally unrelated) to 50 (totally related). Baroni et al. [12] reports the highest performance on this collection using *Word2Vec*.

³¹<http://clic.cimec.unitn.it/elia.bruni/MEN.html>

4.3.1.2 Experimental Setup

We followed experimental setup similar to Baroni et al. [12]. Basically, we implemented two sets of experiments. First, we perform a grid search over MSA’s parameter space to obtain the maximum performing combination of parameters on each dataset. Second, we evaluate MSA in a more realistic settings where we use one of the datasets as a development set for tuning MSA’s parameters and then use the tuned parameters to evaluate MSA’s performance on the other datasets. Some parameters are fixed with all datasets; namely we set consequences size $|Y|=1$, support count $\sigma=1$, and minimum confidence $v=0.0$.

We built the search index using *Wikipedia* dump of August 2016³². The total uncompressed XML dump size was about 52GB representing about 7 million articles. We extracted the articles plain text discarding images and tables. We also discard *References* and *External links* sections (if any). We pruned both articles not under the main namespace and pruned all redirect pages as well. Eventually, our index contained about 4.8 million documents in total.

4.3.1.3 Evaluation

We report the results by measuring the correlation between MSA’s computed relatedness scores and the gold standard provided by human judgments. As in previous studies, we report both Pearson correlation (r) [81] and Spearman rank-order correlation (ρ) [222].

We compare our results with those obtained from three types of semantic repre-

³²<http://dumps.wikimedia.org/backup-index.html>

Table 12: MSA’s Pearson (r) scores on benchmark datasets vs. other techniques for measuring lexical semantic similarity. (\star) from Hassan and Mihalcea [77], (\triangleright) from Baroni et al. [12] predict vectors, (\diamond) from Camacho-Collados et al. [23]. Best performance (bold), second best (underlined)

	<i>MC</i>	<i>RG</i>	<i>WSS</i>	<i>WSR</i>	<i>WS</i>
<i>LSA</i> \star	0.73	0.64	–	–	0.56
<i>ESA</i> \diamond	0.74	0.72	0.45	–	0.49 \star
<i>SSA</i> $_s$ \star	0.87	0.85	–	–	0.62
<i>SSA</i> $_c$ \star	0.88	0.86	–	–	0.59
<i>ADW</i> \diamond	0.79	0.91	0.72	–	–
<i>NASARI</i> \diamond	0.91	0.91	0.74	–	–
<i>Word2Vec</i> \triangleright	0.82	0.84	0.76	0.65	0.68
<i>MSA</i>	0.91	<u>0.87</u>	0.77	0.66	0.69

sensation models. First, statistical co-occurrence models such as LSA [105], CW and BOW [1], and ADW [155]. Second, neural network models like Collobert and Weston (CW) vectors [35], Word2Vec [12], and GloVe [154]. Third, explicit semantics models like ESA [56], SSA [77], and NASARI [23].

4.3.1.4 Results

We report MSA’s correlation scores compared to other models in Tables 12, 13, and 14. Some models do not report their correlation scores on all datasets, so we leave them blank. MSA (last row) represents scores obtained by using *WS* as a development set for tuning MSA’s parameters and evaluating performance on the other datasets using the tuned parameters. The parameter values obtained by tuning on *WS* were: article length $L = 5k$, outdegree $O = 1$, number of concepts $M = 800$, title length $\tau = 2, 3$ for C_s , C_p respectively, and finally minimum rule support $\epsilon = 1$.

Table 12 shows MSA’s Pearson correlation (r) on five benchmark datasets compared to prior work. For Word2Vec, we obtained Baroni et al. [12] predict vectors³³ and used

³³Using <http://clic.cimec.unitn.it/composes/semantic-vectors.html>

Table 13: MSA’s Spearman (ρ) scores on benchmark datasets vs. other techniques for measuring lexical semantic similarity. (\star) from Hassan and Mihalcea [77], (\ddagger) from Hassan and Mihalcea [77], Pilehvar and Navigli [155], (Υ) from Agirre et al. [1], (\S) using pairwise similarities from Camacho-Collados et al. [23], (\diamond) from Pilehvar and Navigli [155], (Ψ) from Pennington et al. [154], (\triangleright) from Baroni et al. [12]. Best performance (bold), second best (underlined)

	<i>MC</i>	<i>RG</i>	<i>WSS</i>	<i>WSR</i>	<i>WS</i>
<i>LSA</i> \star	0.66	0.61	–	–	0.58
<i>ESA</i> \ddagger	0.70	0.75	0.53	–	<u>0.75</u>
<i>SSA</i> $_s$ \star	0.81	0.83	–	–	0.63
<i>SSA</i> $_c$ \star	0.84	0.83	–	–	0.60
<i>CW</i> Υ	–	<u>0.89</u>	0.77	0.46	0.60
<i>BOW</i> Υ	–	0.81	0.70	0.62	0.65
<i>NASARI</i> \S	0.80	0.78	0.73	–	–
<i>ADW</i> \diamond	0.90 ³⁴	0.92	0.75	–	–
<i>GloVe</i> Ψ	0.84	0.83	–	–	0.76
Word2Vec \triangleright	0.82 ³³	0.84	0.76	0.64	0.71
MSA	<u>0.87</u>	0.86	0.77	0.71	0.73

them to calculate Pearson correlation scores. It is clear that, in absolute numbers, MSA consistently gives the highest correlation scores on all datasets compared to other methods except on *RG* where NASARI and ADW [23] performed better.

The best performance, in terms of Pearson correlation, obtained by performing grid search over MSA’s parameter space was 0.97 on *MC*, 0.90 on *RG*, 0.78 on *WSS*, 0.67 on *WSR*, and 0.69 on *WS*.

Table 13 shows MSA’s Spearman correlation scores compared to prior models on same datasets as in Table 12. As we can see, MSA gives highest scores on *WSS* and *WSR* datasets. It comes second on *MC*, third on *RG* and *WS*. We can notice that MSA’s concept enrichments participated in performance gains compared to other explicit concept space models such as ESA and SSA. In addition, MSA *consistently*

³⁴Pairwise similarity scores obtained by contacting authors of Pilehvar and Navigli [155]

Table 14: MSA’s Spearman scores on *MEN* dataset vs. other techniques. (★) from Hill et al. [80], (▷) from Baroni et al. [12]. Best performance (bold), second best (underlined).

	MEN
Skip-gram★	0.44
CW★	0.60
Glove★	0.71
Word2Vec▷	0.79
MSA	<u>0.75</u>

outperformed the popular Word2Vec model on all datasets.

The best performance, in terms of Spearman correlation, obtained by performing grid search of MSA’s parameter space was 0.95 on *MC*, 0.91 on *RG*, 0.78 on *WSS*, 0.72 on *WSR*, and 0.73 on *WS*.

Table 14 shows MSA’s Spearman correlation score vs. other models on *MEN* dataset. As we can see, MSA comes second after Word2Vec giving higher correlation than skip-gram, CW, and GloVe. The results on this dataset prove that MSA is a very advantageous method for evaluating lexical semantic relatedness compared to the popular neural learning models. On another hand, MSA’s Pearson correlation score on *MEN* dataset was 0.73.

We can notice from the results in Tables 12 and Table 13 that measuring semantic relatedness is more difficult than measuring semantic similarity. This is clear from the drop in correlation scores of the relatedness only dataset (*WSR*) compared to the similarity only datasets (*MC*, *RG*, *WSS*). This pattern is common among MSA and all the other techniques which report on these datasets.

4.3.2 Short Text Similarity

4.3.2.1 Dataset

We evaluate MSA’s performance for scoring pairwise short text similarity on Lee50 dataset³⁵ [108]. The dataset contains 50 short documents collected from the Australian Broadcasting Corporation’s news mail service. On average each document has about 81 words. Human annotators were asked to score the semantic similarity of each document to all other documents in the collection. As in previous work, we averaged all human similarity ratings for the same document pair to obtain single score for each pair. This resulted in 1225 unique scored pairs.

4.3.2.2 Experimental Setup

We followed experimental setup similar to Song and Roth [181, 182] for fair comparison. Specifically we created Wikipedia index using August 2016 dump³². We indexed all articles whose length is at least 500 words ($L = 500$) and has at least 30 outgoing links ($O = 30$) using the code base of dataless text classification³⁶. As previously we set consequences size $|Y| = 1$ and minimum confidence $v = 0.0$. We also set support count $\sigma = 5$ and relax title length τ .

4.3.2.3 Evaluation

We report the both Pearson (r) and Spearman (ρ) correlations between MSA’s similarity scores and human judgments using a concept vector of size 500 ($M = 500$) as in Song and Roth [182]. We compare our results to ESA’s results using Song and Roth

³⁵<http://faculty.sites.uci.edu/mdlee/similarity-data/>

³⁶http://cogcomp.cs.illinois.edu/page/software_view/DatalessHC

Table 15: MSA’s Spearman (ρ) and Pearson (r) scores on Lee50 dataset vs. other techniques. (\star) from [77].

	<i>LSA</i> \star	<i>SSA</i> \star	<i>ESA</i> ³⁶	MSA
<i>Spearman</i> (ρ)	0.46	0.49	0.61	0.62
<i>Pearson</i> (r)	0.70	0.68	0.73	0.75

[180] implementation. We also compare our results to other semantic representation models such as LSA and SSA.

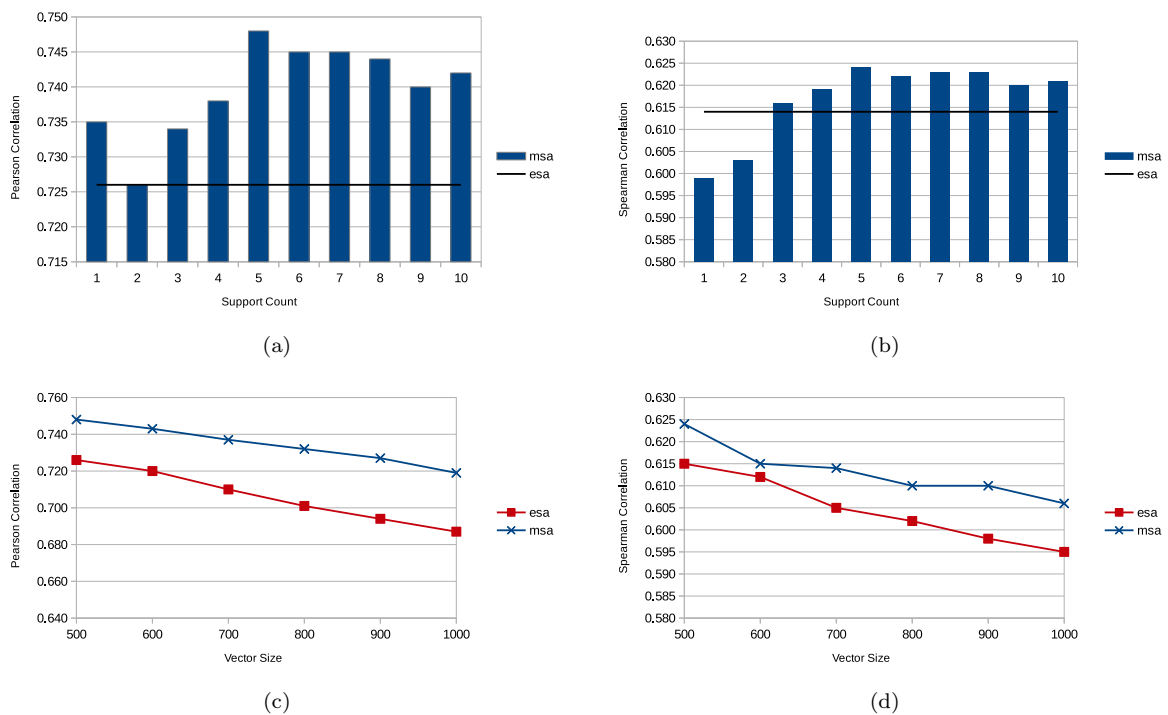


Figure 11: MSA’s correlation scores on Lee50 dataset. (a) Pearson (r) correlation when varying support count, (b) Spearman (ρ) correlation when varying support count, (c) Pearson (r) correlation when varying vector size count, and (d) Spearman (ρ) correlation when varying vector size.

4.3.2.4 Results

As we can see in Table 15, MSA outperforms prior models on Lee50 dataset. ESA comes second after MSA which shows the potential of concept space augmentation using implicitly associated concepts discovered through MSA’s concept-concept asso-

ciation learning.

We performed another experiment in order to assess the impact of parameter tuning on the reported results in Table 15. Figure 11-a and 11-b show MSA’s Pearson (r) and Spearman (ρ) correlation scores when varying the support count parameter (σ) within the 1 to 10 range in steps of 1. As we can see MSA’s r correlation scores are consistently higher than ESA. In addition, ρ correlation scores of MSA are higher than ESA score for all σ values between 3 and 10. Figure 11-c and 11-d show MSA’s r and ρ correlation scores compared to ESA when varying the vector size parameter (M) within the 500 to 1000 range in steps of 100. As we can see both r and ρ correlation scores of MSA are consistently higher than ESA.

Two other observations we can notice from Figure 11-c and 11-d. First, both r and ρ scores of MSA and ESA tend to decrease as we increase the vector size. We believe this is because more irrelevant concepts are added to the concept vector causing divergence from a ”better” to a ”worse” representation of the given document. Second, and more importantly, MSA’s r and ρ correlation scores are higher than SSA and LSA over all values of σ and M for both r and ρ correlations. This experiment reflects that MSA’s semantic representations are largely robust against parameters tuning.

4.4 A Study on Statistical Significance

Through the results section, we kept away from naming state-of-the-art method. That was due two facts. First, the differences between reported correlation scores were small. Second, the size of the datasets was not that large to accommodate for such

Table 16: Steiger’s Z significance test on the differences between Spearman correlations (ρ) using 1-tailed test and 0.05 statistical significance. (\triangleright) using Baroni et al. [12] predict vectors, (\star) using Camacho-Collados et al. [23] pairwise similarity scores, (\diamond) using Pilehvar and Navigli [155] pairwise similarity scores.

	<i>MC</i>		<i>RG</i>		<i>WSS</i>		<i>WSR</i>		<i>WS</i>		<i>MEN</i>	
	ρ	<i>p</i> -value	ρ	<i>p</i> -value	ρ	<i>p</i> -value	ρ	<i>p</i> -value	ρ	<i>p</i> -value	ρ	<i>p</i> -value
	<i>MSA_t</i>											
<i>Word2Vec_t</i> ^{\triangleright}	0.84	0.168	0.78	0.297	0.79	0.357	0.70	0.019	0.72	0.218	0.78	0.001
NASARI ^{\star}	0.73	0.138	0.77	0.030	0.70	0.109	–	–	–	–	–	–
ADW ^{\diamond}	0.78	0.258	0.78	0.019	0.67	0.271	–	–	–	–	–	–
	<i>ADW^{\diamond}</i>											
<i>Word2Vec_t</i> ^{\triangleright}	0.80	0.058	0.81	0.003	0.68	0.5	–	–	–	–	–	–
NASARI ^{\star}	0.82	0.025	0.80	0.0	0.76	0.256	–	–	–	–	–	–
	<i>Word2Vec_t</i> ^{\triangleright}											
NASARI ^{\star}	0.75	0.387	0.71	0.105	0.66	0.192	–	–	–	–	–	–

small differences. These two facts raise a question about the *statistical significance* of improvement reported by some method A compared to another well performing method B.

We hypothesize that the best method is not necessarily the one that gives the highest correlation score. In other words, being state-of-the-art does not require giving the highest correlation, rather giving a relatively high score that makes the difference with any other higher score statistically insignificant.

To test our hypothesis, we decided to perform statistical significance tests on the top reported correlations. Initially we targeted Word2Vec, GloVe, ADW, and NASARI besides MSA. We contacted several authors and some of them thankfully provided us with pairwise relatedness scores on the corresponding benchmark datasets. We also utilized the publicly available semantic vectors of some models like Baroni et al. [12] predict vectors.

To measure statistical significance, we performed Steiger’s Z significance test [188]. The purpose of this test is to evaluate whether the difference between two dependent

correlations obtained from the same sample is statistically significant or not, i.e., whether the two correlations are statistically equivalent.

Steiger’s Z test requires to calculate the correlation between the two correlations. We applied the tests on Spearman correlations (ρ) as it is more commonly used than Pearson (r) correlation. We conducted the tests using correlation scores of MSA’s tuned model on *WS* dataset, Word2Vec, ADW, and *NSASRI*.

Table 16, shows the results using 1-tailed test with significance level 0.05³⁷. For each dataset, we report method-method Spearman correlation (ρ) calculated using reported scores in Table 13 and Table 14. We report p -value of the test as well.

On *MC* dataset, the difference between MSA score and all other methods was statistically insignificant. Only ADW score was statistically significant compared to NSASARI. This implies that MSA can be considered statistically a top performer on *MC* dataset.

On *RG* dataset, MSA gave significant improvement over NASARI. ADW score was significantly better than Word2Vec, NASARI, and MSA. Overall, ADW can be considered the best on *RG* dataset followed by MSA and Word2Vec (their ρ scores are 0.92, 0.86, and 0.84 respectively).

On *WSS*, though MSA achieved the highest score ($\rho=0.77$), no significant improvement was proved. Therefore, the differences between the four methods can be considered statistically insignificant.

On *WSR*, *WS*, and *MEN* datasets, we could obtain pairwise relatedness scores of

³⁷Using 2-tailed test rather than 1-tailed test would double all the p -value scores in Table 16, subsequently increasing the confidence in the Null hypothesis which supports our argument that the differences are not statistically significant.

Word2Vec only. The significance test results indicated that, the improvement of MSA over Word2Vec on *WS* was statistically insignificant (their ρ scores are 0.77 and 0.76 respectively). On the other hand, MSA was statistically better than Word2Vec on *WSR* dataset (their ρ scores are 0.71 and 0.64 respectively), while Word2Vec was statistically better than MSA on *MEN* dataset (their ρ scores are 0.79 and 0.75 respectively).

This comparative study is one of the main contributions of this thesis. To our knowledge, this is the first study that addresses evaluating the statistical significance of results across various semantic relatedness methods. Additionally, this study positions MSA as a state-of-the-art method for measuring semantic relatedness compared to other explicit concept-based representation methods such as ESA and SSA. It also shows that MSA is very competitive to other neural-based representations such as Word2Vec and GloVe.

4.5 Conclusion

In this chapter, we presented MSA, a novel approach for semantic analysis which employs data mining techniques to create conceptual vector representations of text. MSA is motivated by inability of prior concept space models to capture implicit relations between concepts. To this end, MSA mines for implicit concept-concept associations through *Wikipedia's* "See also" link graph.

Intuitively, "See also" links represent related concepts that might complement the conceptual knowledge about a given concept. Furthermore, it is common in most online encyclopedic portals to have a "See also" or "Related Entries" sections opening

the door for more conceptual knowledge augmentation using these resources in the future.

Through empirical results, we demonstrated MSA’s effectiveness to measure lexical semantic relatedness on benchmark datasets. In absolute numbers, MSA could consistently produce higher Pearson correlation scores than other explicit concept space models (ESA, SSA) on all data sets. Additionally, MSA could produce higher scores than ADW and NASARI on four out of five datasets. On another hand, MSA scores were higher than predictive models built using neural networks (e.g., Word2Vec).

Regarding Spearman correlation, MSA produced the highest scores on two datasets (*WSS* and *WSR*). Results on other datasets were very competitive in absolute numbers. Specifically, MSA gave higher Spearman correlations than Glove and Word2Vec on both *MC* and *RG* datasets. Additionally, MSA gave higher correlation on *MEN* dataset than neural-based representations including skip-gram, CW, and Glove.

The results show MSA competitiveness compared to state-of-the-art methods. More importantly, our method produced significantly higher correlation scores than previous explicit semantics methods (ESA and SSA). The good performance demonstrates the potential of MSA for augmenting the explicit concept space by other semantically related concepts which contribute to understanding the given text.

In this chapter, we introduced the first comparative study which evaluates the statistical significance of results from across top performing semantic relatedness methods. We used Steiger’s *Z* significance test to evaluate whether reported correlations from two different methods are statistically equivalent even if they are numerically different. We believe this study will help the research community to better evalu-

ate and position state-of-the-art techniques at different application areas. The study proved that, statistically, MSA results are either better than or equivalent to state-of-the-art methods on all datasets except *RG* where ADW was better, and *MEN* where Word2Vec was better.

MSA is a general purpose semantic representation approach which builds explicit conceptual representations of textual data. We argue that the expressiveness and interpretability of MSA representation make it easier for humans to manipulate and interact with. These two advantages favors MSA over the popular Word2Vec representation. As we will show in Chapter 5, MSA could be used in many text understanding applications which require interactivity and visualization of the underlying representation such as interactive semantic search, concept tracking, technology landscape analysis.

MSA is an efficient technique because it employs an inverted search index to retrieve semantically related concepts to a given text. Additionally, mining for concept-concept association rules is done offline making it scalable to huge amounts of data.

The approach presented in this chapter serves directly our overall goal exploiting KBs in order to increase the effectiveness and expressiveness of semantic representations of text structures. We showed through empirical results that the concept-based representation of MSA is more effective (in terms of performance) than other semantic representation models such as ESA, SSA, and LSA on benchmark datasets for measuring word and short text semantic associations.

CHAPTER 5: INNOVATION ANALYTICS USING MINED SEMANTIC ANALYSIS

In this chapter, we introduce our first steps toward building a semantic-driven interactive and visual framework powered by Mined Semantic Analysis (MSA). We analyze the applicability of such framework to innovations and patents³⁸ analytics. Our framework provides cognitive assistance to its users through a Web-based visual and interactive interface. We demonstrate applying the acquired knowledge from MSA representations to support many cognition and knowledge-based use cases for innovation analysis including technology exploration and landscaping, competitive analysis, prior art search and others.

The work presented in this chapter relates to our goal improving non-expert users' usability of the semantic representations through visualization and interactivity.

5.1 Background and Motivation

Patents and innovations represent *proxies* for economic, technological, and even social activities. Therefore, patent analysis has received considerable attention in the literature³⁹ [48, 220, 37, 131, 118, 98]. Typical innovation management use cases include:

1. **Technology exploration** in order to capture new and trendy technologies in a specific domain and subsequently using them to create ideas for new innovative

³⁸We use innovations and patents interchangeably throughout the chapter

³⁹<http://users.cis.fiu.edu/lzhan015/patmining.html>

services.

2. **Technology landscape analysis** in order to assess the density of patent filings of specific technology and subsequently direct R&D activities accordingly.
3. **Competitive analysis and benchmarking** in order to identify strengths and differences of corporate's own patent portfolio compared to other key players working on related technologies.
4. **Patent ranking and scoring** in order to quantify the strength of the claims of an existing or a new patent.
5. **Prior art search** in order to retrieve patent documents and other scientific publications relevant to a new patent application.

All those innovation management activities require tremendous level of *domain expertise* which, even if available, must be integrated with highly sophisticated and intelligent analytics that provide cognitive and *interactive assistance* to the users.

Due its technical nature, patent language tends to be highly sophisticated with *complex vocabulary, legal jargon, and domain specific terminology*. Most research in automated patent analysis is inspired by either *content-based* (e.g., term co-occurrence) or *metadata-based* (e.g., bibliographic data) methods.

We, alternatively, embrace *semantic-driven analysis* of innovation data. Our hypothesis is that, by subtle incorporation of *external conceptual knowledge*, we could bridge the linguistic and domain expertise gaps and provide non-expert users *cognitive assistance* that would not be achievable by using the limited content-based

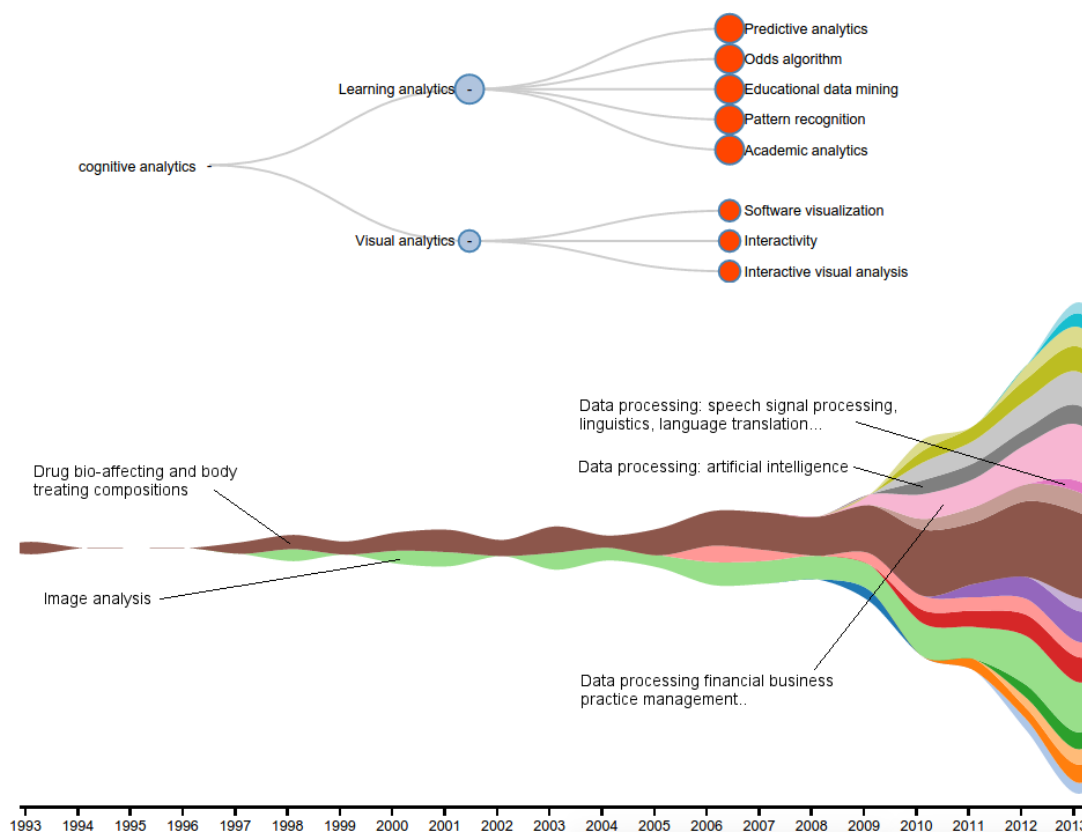


Figure 12: Top: Concept graph of *Cognitive Analytics*; explicit concepts are light blue nodes, and implicit concepts are red nodes. Bottom: ThemeRiver plot showing patenting evolution of *Cognitive Analytics* and related technologies in its concept graph.

approaches.

To this end, we propose a *Web-based semantic framework for innovation analytics* to demonstrate typical use cases which map to real-world cognitive tasks that practitioners deal with today. We employ MSA, our novel semantic representation approach which employs data mining techniques. As we showed in Chapter 4, MSA, given an input text, constructs a conceptual knowledge graph whose nodes are encyclopedic concepts and links are quantified associations between those concepts. MSA builds that knowledge graph offline by mining for concept-concept associations in a target encyclopedic textual corpus (e.g., *Wikipedia*) using association rules mining [3]. Once

constructed, this rich concept graph can be used for several tasks like semantic search, concept expansion, measuring semantic relatedness, word sense disambiguation, resolving vocabulary mismatch, and others.

5.2 Case Study

In this section we demonstrate various patent analytics scenarios using our semantic-driven visual framework powered by MSA.

5.2.1 Technology Exploration and Landscape Analysis

Consider "*Cognitive Analytics*" as an example technology. Figure 12 (top) shows the concept graph of "*Cognitive Analytics*" by retrieving the top 10 most semantically related concepts using MSA (node size reflects association strength). We can clearly notice that: 1) those concepts are very associated with "*Cognitive Analytics*" as well as with one another, and 2) they cover a wide spectrum of the technological and conceptual landscape.

All patent data are indexed into our framework; consequently, we could facilitate landscape analysis of "*Cognitive Analytics*" related technologies by showing patenting and innovation progression as the ThemeRiver shown in Figure 12 (bottom). Each stream represents a patent class where stream width is proportional to the number of granted patents per class over 20 years between 1993 and 2013. Streams allow practitioners to capture patenting trends that would otherwise require investigating huge number of patent documents in an iterative and time consuming manner. Those trends include:

- Patents were limited to two classes until 2005; "*(382) image analysis*" (light

green), and "*(514) drug bio-affecting and body treating compositions*" (dark brown).

- By 2010, the granted patents expanded to cover various patent classes; e.g., "*(706) data processing: artificial intelligence*" (dark gray) and "*(705) data processing financial business practice management...*" (pink).
- A new application domain of "*Cognitive Analytics*" and related technologies emerged in 2013; "*(706) data processing: speech signal processing, linguistics, language translation...*" (purple).

Our framework encourages *human-in-the-loop* processing by providing users with interactive visualizations that support their cognitive tasks. For example, users can easily interact with the visualizations provided in Figure 12 by controlling the *concept association strength*, *pruning* possibly irrelevant concepts, and *zooming* in each stream to navigate through individual patents under that stream.

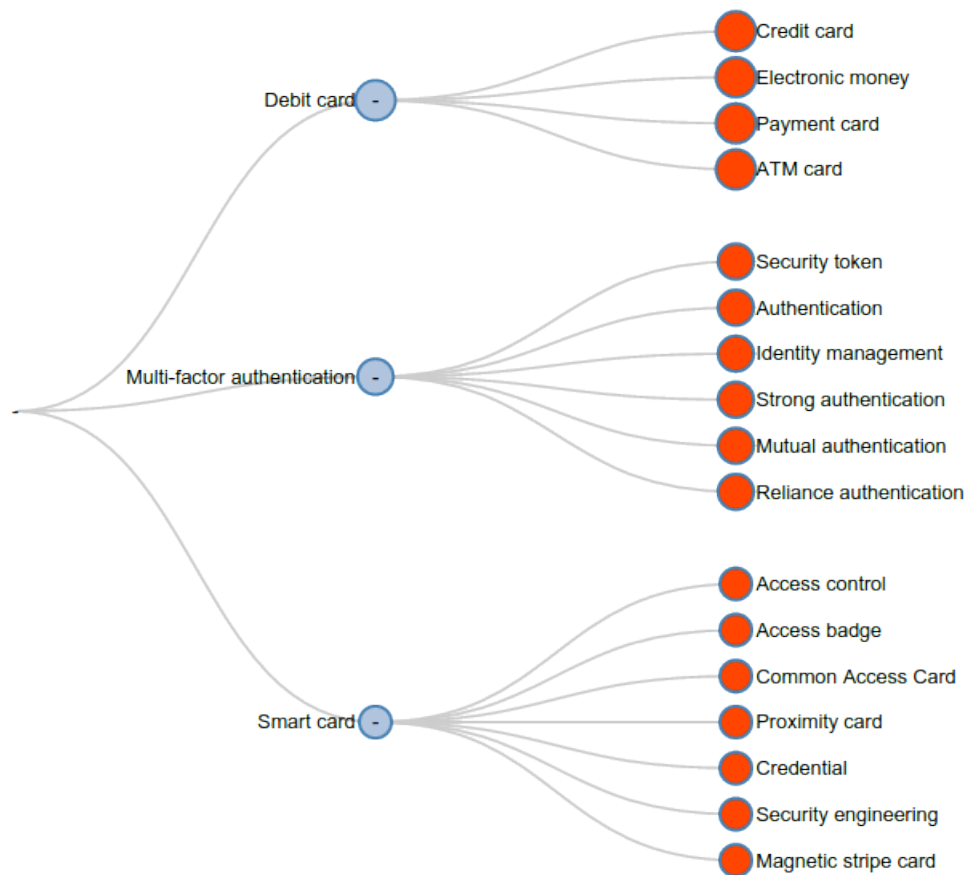
5.2.2 Competitive Intelligence

Another application of our semantic-driven framework in the innovations analytics domain is competitive intelligence. This use case explains how MSA can be applied to: 1) define the intellectual property portfolio of an organization, and 2) identify other key players with similar portfolios which could be candidates for acquisition.

As pointed earlier, working with patents language poses many challenges. The *vocabulary mismatch* problem is one of the prominent problems in computational linguistics generally and patents analysis specifically. To evince their work novelty, patent authors deliberately use different vocabulary to refer to the same concepts.

Table 17: *Bank of America's* sample patent titles.

No	Publication #	Title
1	US 8745155	Network storage device collector
2	US 8719160	Processing payment items
3	US 8600882	Prepaid card budgeting
4	US 8444051	Self-Service machine problem code
5	US 8301530	Automatic savings program
6	US 8136148	reusable authentication experience tool
7	US 8005728	Currency ordering by denomination
8	US 7982604	tamper-indicating monetary package
9	US 8635159	Self-service terminal limited access personal identification number
10	US 8634322	Apparatus and methods for adaptive network throttling

Figure 13: Concept graph of Bank of America (*BofA*)'s 100 patent titles. Light blue nodes are explicit concepts and red nodes are implicit ones.

Resolving similarity and relatedness between technical concepts is very important in many innovation analysis applications such as concept expansion and tracking, monitoring technology evolution, technology to industry mappings, and others.

To define the portfolio of an organization, we built a big index of all US granted patents⁴⁰ between 1976 and Oct. 2014. We used Apache Solr to build and search the index. The total index size was about 200GB comprising around 4.7 million documents. For each patent, we indexed its *title*, *abstract*, *description*, *claims*, *assignee*, and *publication date*.

The competitive intelligence scenario starts with a seed organization and ends with potential key players with similar portfolios. In the process, target organization's portfolio is defined in terms of technological and technical concepts expressed explicitly or implicitly in the organization's patents.

We exemplify by considering Bank of America⁴¹ (*BofA*) as a target organization. By searching our patents index, we found approximately 790 patents whose assignee is *BofA*. Portfolio identification is a multi-step process that uses representative description of the patents; e.g. their *titles*, *abstracts*, *descriptions*, and/or *claims*. We extract *titles* of 100 patents at random (see Table 17 for sample titles) as a representative description of *BofA*'s innovations. Then, we pass all titles as a single snippet to MSA to discover the corresponding concept representation.

Figure 13 shows MSA's top 20 relevant concepts which represent *BofA*'s portfolio using titles from Table 17. As we can notice, those concepts are semantically related to the titles in Table 17.

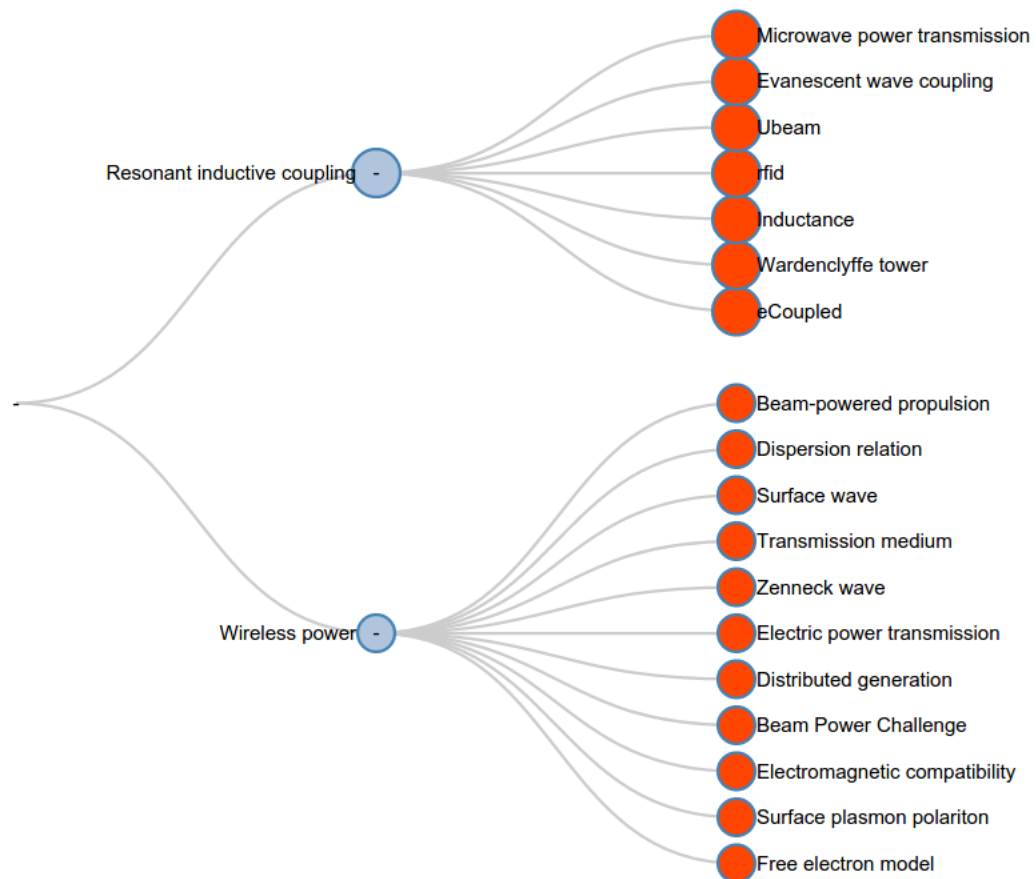
The final step in our competitive analysis scenario is to identify key players with similar portfolios to *BofA*. To do this step, we take the top ranked concepts and com-

⁴⁰<https://data.uspto.gov/uspto.html>

⁴¹<https://www.bankofamerica.com/>

Table 18: *Witricity's* sample patent titles.

No	Publication #	Title
1	US 8106539	Wireless energy transfer for refrigerator application
2	US 8618696	Wireless energy transfer systems
3	US 8497601	Wireless energy transfer converters
4	US 8569914	Wireless energy transfer using object positioning for improved k
5	US D709855	Clock radio phone charger
6	US D705745	Printed resonator coil
7	US 8471410	Wireless energy transfer over distance using field shaping to improve the coupling factor
8	US D692010	Wireless power source
9	US 8729737	Wireless energy transfer using repeater resonators
10	US 8805530	Power generation for implantable devices

Figure 14: Concept graph of *Witricity's* 10 patent titles. Light blue nodes are explicit concepts and red nodes are implicit ones.

bine them to construct a search query against the patents index. We limit the search to patent *claims* as they define in technical terms the scope of protection sought by the inventor. Among the top ranked key players in the competitors list were com-

panies like *ActivIdentity*⁴² which is specialized in identity assurance, *SecureEnvoy*⁴³ which is specialized in authentication and verification, and *IBM*.

To validate the robustness of our semantic framework, we repeated the same competitive analysis experiment on *Witricity*⁴⁴, which is specialized in wireless energy transfer using resonant magnetic coupling. Table 18 shows titles of 10 *Witricity*'s patents. Figure 14 shows the top 20 relevant concepts representing *Witricity*'s portfolio using those titles. We validated the relevance of the retrieved concepts to the wireless energy industry based on feedback from a domain expert. To close the loop, we retrieved similar key players to *Witricity* and the list included *Qualcomm*⁴⁵, *Powermat*⁴⁶, and *Mojo Mobility*⁴⁷ which all provide wireless charging solutions.

5.3 Conclusion

In this chapter, we presented a semantic, visual, and interactive framework for innovation analytics powered by MSA. The framework supports various cognition and knowledge intensive tasks of patent analysis and thus maximizes the potential for user exploration. Specifically, we demonstrated a case study using MSA's semantic-driven framework for technology landscape analysis and competitive intelligence. This work goes in harmony with our overall objective to increase the usability of the semantic representation allowing users to better explore, analyze, and get insights from unstructured texts.

⁴²<http://portal.actividentity.com/>

⁴³<https://www.secureenvoy.com/>

⁴⁴<http://witricity.com/>

⁴⁵<https://www.qualcomm.com/>

⁴⁶<http://www.powernmat.com/>

⁴⁷<http://www.mojomobility.com/home>

CHAPTER 6: LEARNING CONCEPT AND ENTITY EMBEDDINGS

In this Chapter we focus on increasing the effectiveness and efficiency of the interpretable explicit concept space representations including MSA. As we highlighted earlier, explicit concept space models have proven efficacy for text representation in many natural language and text mining applications. The idea is to embed textual structures into a semantic space of concepts which captures the main ideas, objects, and the characteristics of these structures. The so called Bag-of-Concepts (BoC) representation suffers from data sparsity causing low similarity scores between similar texts due to low concept overlap. To address this problem, we propose two neural embedding models to learn continuous⁴⁸ concept vectors. Once they are learned, we propose an efficient vector aggregation method to generate fully continuous BoC representations. We evaluate our concept embedding models on three tasks: 1) measuring entity semantic relatedness and ranking where we achieve 1.6% improvement in correlation scores, 2) dataless concept categorization where we achieve state-of-the-art performance and reduce the categorization error rate by more than 5% compared to five prior word and entity embedding models, and 3) dataless document classification where our models outperform the sparse BoC representations. In addition, by exploiting our efficient linear time vector aggregation method, we achieve better

⁴⁸We use the terms continuous, dense, distributed vectors interchangeably to refer to real-valued vectors.

accuracy scores with much less concept dimensions compared to previous BoC densification methods which operate in polynomial time and require hundreds of dimensions in the BoC representation.

Table 19: Top 3 concepts generated using ESA [56] for two 20-newsgroups categories (Hockey and Guns) along with top 3 concepts of sample instances. Using exact match similarity scoring (as in ESA) result in low scores between similar instance and category concept vectors. When using concept embeddings (our models), we obtain relatively higher and more representative similarities.

Category	Top 3 Concepts	Instance Top 3 Concepts	ESA	CCX	CRX
Hockey	<ul style="list-style-type: none"> - Detroit Red Wings, - History of the Detroit Red Wings, - History of the NHL on United States television 	Instance (53798) <ul style="list-style-type: none"> - History of the Detroit Red Wings, - Detroit Red Wings, - Pittsburgh Penguins 	0.73	0.95	0.95
		Instance (54551) <ul style="list-style-type: none"> - Paul Kariya, - Boston Bruins, - Bobby Orr 	0.0	0.84	0.80
Guns	<ul style="list-style-type: none"> - Waco siege, - Overview of gun laws by nation, - Gun violence in the United States 	Instance (54387) <ul style="list-style-type: none"> - Overview of gun laws by nation, - Waco siege, - Gun politics in the United States 	0.71	0.94	0.93
		Instance (54477) <ul style="list-style-type: none"> - Concealed carry in the United States, - Overview of gun laws by nation, - Gun laws in California 	0.33	0.80	0.75

6.1 Background & Motivation

As mentioned in Chapter 4, explicit concept space models utilize concept vectors (aka Bag-of-Concepts (BoC)) as the underlying semantic representation of a given text through a process called *conceptualization*, which is mapping the text into relevant concepts capturing its main ideas, objects, and their characteristics.

Similar to the traditional Bag-of-Words (BoW) representation, the BoC vector is

a multidimensional *sparse* vector whose dimensionality is the same as the number of concepts in the employed Knowledge Base (KB) (typically *millions*). Consequently, it suffers from *data sparsity* causing low similarity scores between similar texts due to low concept overlap. Moreover, the BoC vector is generated from the top n concepts which have relatively high association scores with the input terms (typically few hundreds). Thus each text snippet is mapped to a very sparse vector of millions of dimensions having only few hundreds nonzero values leading to the *BoC sparsity problem* [153].

Having such sparse representation and using exact match similarity scoring measure, we can expect that two very similar text snippets might have *zero similarity* score if they map to *different but very related set of concepts* [182]. We demonstrate this fact in Table 19 (ESA column).

In this Chapter we utilize *neural-based representations* to overcome the BoC sparsity problem. The basic idea is to *map* each concept to a *fixed size continuous vector*. These vectors can then be used to compute concept-concept similarity and thus overcome the concept mismatch problem.

Our work is also motivated by the success of recent neural-based methods for learning word embeddings in capturing both syntactic and semantic regularities using simple vector arithmetic [138, 139, 154]. For example, inferring analogical relationships between words: $vec(king) - vec(man) + vec(woman) = vec(queen)$. This indicates that the learned vectors encode meaningful multi-clustering for each word.

However, word vectors suffer from significant limitations. First, each word is assumed to have a *single meaning* regardless of its context and thus is represented by a

single vector in the semantic space (e.g., *charlotte (city)* vs. *charlotte (given name)*). Second, the space contains vectors of single words only. Vectors of multiword expressions (MWEs) are typically obtained by averaging the vectors of individual words. This often produces inaccurate representations especially if the meaning of the MWE is different from the composition of meanings of its individual words (e.g., *vec(north carolina)* vs. *vec(north)+vec(carolina)*). Additionally, mentions that are used to refer to the same concept would have different embeddings (e.g., *u.s.*, *america*, *usa*), and the model might not be able to place those individual vectors in the same sub-cluster, especially the rare surface forms.

We propose two *neural embedding* models in order to learn continuous concept vectors based on the skip-gram model [139]. Our first model is the *Concept Raw Context* model (CRX) which utilizes raw concept mentions in a large scale textual KB to jointly learn embeddings of both words and concepts. Our second model is the *Concept-Concept Context* model (CCX) which learns the embeddings of concepts from their conceptual contexts (i.e., contexts containing surrounding concepts only). After learning the concept vectors, we propose an *efficient BoC aggregation* method. We perform *weighted average* of the individual concept vectors to generate fully *continuous* BoC representations (CBoC). This aggregation method allows measuring the similarity between pairs of BoC in *linear time* which is more efficient than previous methods that require *quadratic* or at least *log-linear* time if optimized (see Equation 5). Our embedding models produce more *representative* similarity scores for BoC containing *different but semantically similar* concepts as shown in Table 19 (columns 2-3).

We evaluate our embedding models on three tasks:

1. An intrinsic task of measuring entity semantic relatedness and ranking where we achieve 1.6% improvement in correlation scores.
2. Dataless concept categorization where we achieve state-of-the-art performance and reduce the categorization error rate by more than 5% compared to five prior word and entity embedding models.
3. An extrinsic task of dataless document classification. Experimental results show that we can achieve better accuracy using our efficient BoC densification method compared to the original sparse BoC representation with much less concept dimensions.

The contributions of our approach are fourfold: First, we propose two *low cost* concept embedding models which learn concept representations from concept mentions in free-text corpora. Our models require few hours rather than days to train. Second, we show through empirical results the efficacy of the learned concept embeddings in measuring entity semantic relatedness and concept categorization. Our models achieve *state-of-the-art performance* on two concept categorization datasets. Third, we propose simple and efficient vector aggregation method to obtain *fully dense BoC in linear time*. Fourth, we demonstrate through experiments on dataless document classification that we can obtain better accuracy using the dense BoC representation with much less dimensions (few in most cases), reducing the *computational cost* of generating the BoC vector significantly.

6.2 Related Work

6.2.1 Text Conceptualization

We highlighted in Chapter 4 the value of text conceptualization as a way of automating the generalizations humans perform while figuring out text meanings. Broadly speaking, conceptualization methods are either vector-based or knowledge-based. Vector-based methods utilize semi-structured KBs such as *Wikipedia* in order to construct the concept space which is defined by all *Wikipedia* article titles. Knowledge-based methods use more structured concept KBs such as *Microsoft Knowledge Graph* (aka *Probase*⁴⁹) [212] which is a probabilistic KB of millions concepts and their relationships. *Probase* uses syntactic patterns in order to mine for concepts and relationships. Despite its effectiveness, the dependency of *Probase* on syntactic patterns can be a limitation especially for languages other than English. In addition, we expect augmenting and maintaining these syntactic patterns to be costly and labor intensive. We argue that concept embeddings allow *simpler and more efficient representations*, simply because similarity scoring between individual or vectors of concepts can be performed using vector arithmetic. While the *Probase* hierarchy allows only symbolic matching, which still suffers from data sparsity. On another hand, we spotted some cases where *Probase* probabilities were atypical⁵⁰. This is due to learning concept categories from a limited set of syntactic patterns which does not cover all concept mention patterns. Concept embeddings relax this requirement by *exploiting all concept mentions* in order to learn the embedding vector and therefore

⁴⁹<https://concept.research.microsoft.com>

⁵⁰ $p(\text{Arabic coffee} \mid \text{Beverage}) = 0$

might be utilized to curate such atypical *Probase* assertions.

6.2.2 Concept/Entity Embeddings

Neural embedding models have been proposed to learn distributed representations of concepts/entities⁵¹. Song and Roth [182] proposed using the popular Word2Vec model [138] to obtain the embeddings of each concept by averaging the vectors of the concept’s individual words. For example, the embeddings of *Microsoft Office* would be obtained by averaging the embeddings of *Microsoft* and *Office* obtained from the Word2Vec model. Clearly, this method disregards the fact that the semantics of multiword concepts is different from the semantics of their individual words.

More robust concept and entity embeddings can be learned from the general knowledge about the concept in encyclopedic KB (e.g., its article) and/or from the structure of a hyperlinked KB (e.g., its link graph). Such concept embedding models were proposed by Hu et al. [85], Li et al. [110], and Yamada et al. [214] who all utilize the skip-gram learning technique [139], but differ in how they define the context of the target concept.

Li et al. [110] extended the embedding model proposed by Hu et al. [85] by jointly learning entity and category embeddings from contexts defined by all other entities in the target entity article as well as its category hierarchy in *Wikipedia*. This method has the advantage of learning embeddings of both entities and categories jointly. However, defining the entity contexts as pairs of the target entity and all other entities appearing in its corresponding article might introduce noisy contexts, especially for

⁵¹In this chapter, we use the terms "concept" and "entity" interchangeably.

long articles. For example, the *Wikipedia* article for "*United States*" contains links to "*Kindergarten*", "*First grade*", and "*Secondary school*" under the "*Education*" section.

Yamada et al. [214] proposed a method based on the skip-gram model to jointly learn embeddings of words and entities using contexts generated from surrounding words of the target entity or word. The authors also proposed incorporating *Wikipedia* link graph by generating contexts from all entities with outgoing link to the target entity to better model entity-entity relatedness.

Our models also learn word and concept embeddings jointly. Mapping both words and concepts into the same semantic space allows us to easily measure word-word, word-concept, and concept-concept semantic similarities. In addition, our CRX model (described in Section 6.3.2) extends the context of each word/concept by including nearby concept mentions and not only nearby words. Therefore, we better model the local contextual information of concepts and words in *Wikipedia*, treated as a textual KB. During training, we generate word-word, word-concept, concept-word, and concept-concept contexts (cf. Equation 15). In Yamada et al. [214] model, concept-concept contexts are generated from *Wikipedia* link graph not from their raw mentions in *Wikipedia* text. In the CCX model, we define concept contexts by all surrounding concepts within a window of fixed size.

Generating contexts from raw text mentions makes our models *not restricted to hyperlinked encyclopedic textual corpora* only. This facilitates exploiting other free-text corpora with annotated concept mentions (e.g., news stories, scientific publications, medical guidelines...etc). Moreover, our proposed models are *computationally less*

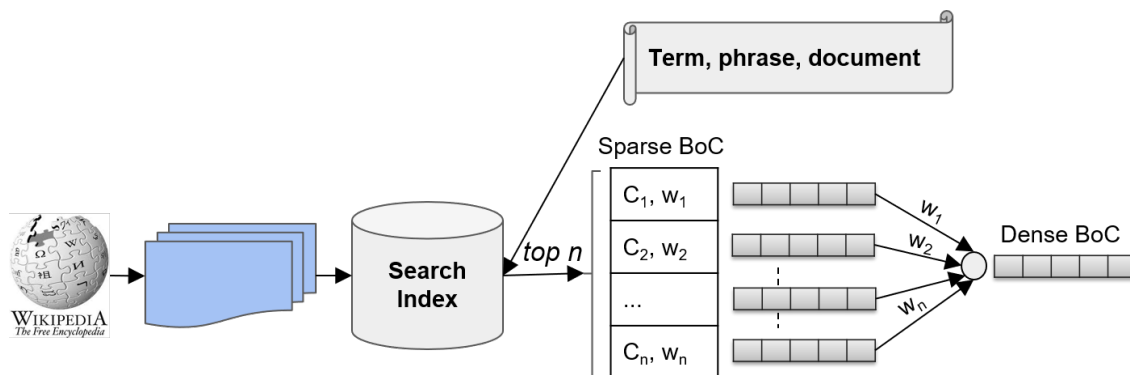


Figure 15: Densification of the bag-of-concepts using weighted average of the learned concept embeddings. The concept space is defined by all Wikipedia article titles. The concept vector is created from the top n hits of searching a *Wikipedia* inverted index with the given text.

costly than Hu et al. [85] and Yamada et al. [214] models as they require few hours rather than days to train on similar computing resources.

6.2.3 Bag-of-Concepts Densification

Densification of the Bag-of-Concepts (BoC) is the process of converting the sparse BoC into a continuous BoC (CBoC) (aka dense BoC) in order to overcome the BoC sparsity problem. The process requires first mapping each concept into a continuous vector using representation learning. Song and Roth [182] proposed three different mechanisms for aligning the concepts at different indices given a sparse BoC pair (\mathbf{u}, \mathbf{v}) in order to increase their similarity score.

The *many-to-many* mechanism works by averaging all pairwise similarities. The *many-to-one* mechanism works by aligning each concept in \mathbf{u} with the most similar concept in \mathbf{v} (i.e., its best match). Clearly, the complexity of these two mechanisms is *quadratic*. The third mechanism is the *one-to-one*. It utilizes the Hungarian method in order to find an optimal alignment on a one-to-one basis [151]. This mechanism per-

formed the best on the task of dataless document classification and was also utilized by Li et al. [110]. However, the Hungarian method is a combinatorial optimization algorithm whose complexity is *polynomial*. Our proposed densification mechanism is more efficient than these three mechanisms as its complexity is *linear* with respect to the number of *nonzero* elements in the BoC. Additionally, it is simpler as it does not require tuning a cutoff threshold for the minimum similarity between two aligned concepts as in previous work. Figure 15 shows a schematic diagram of our efficient densification mechanism applied to a BoC generated from a *Wikipedia* inverted index. We simply perform weighted average of the individual concept vectors in the obtained BoC.

6.3 Learning Concept Embeddings

A main objective of learning concept embeddings is to overcome the inherent problem of *data sparsity* associated with the BoC representation. Here we try to learn continuous concept vectors by building upon the skip-gram embedding model [139].

6.3.1 Skip-gram

In the conventional skip-gram model, a set of contexts are generated by sliding a context window of predefined size over sentences of a given text corpus. Vector representation of a target word is learned with the objective to maximize the ability of predicting surrounding words of that target word.

Formally, given a training corpus of V words w_1, w_2, \dots, w_V . The skip-gram model

aims to maximize the average log probability:

$$\frac{1}{V} \sum_{i=1}^V \sum_{-s \leq j \leq s, j \neq 0} \log p(w_{i+j}|w_i) \quad (13)$$

where s is the context window size, w_i is the target word, and w_{i+j} is a surrounding context word. The softmax function is used to estimate the probability $p(w_O|w_I)$ as follows:

$$p(w_O|w_I) = \frac{\exp(\mathbf{v}_{w_O}^\top \mathbf{u}_{w_I})}{\sum_{w=1}^W \exp(\mathbf{v}_w^\top \mathbf{u}_{w_I})} \quad (14)$$

where \mathbf{u}_{w_I} and \mathbf{v}_{w_O} are the input and output vectors respectively and W is the vocabulary size. Mikolov et al. [139] proposed hierarchical softmax and negative sampling as efficient alternatives to approximate the softmax function which becomes computationally intractable when W becomes huge.

Our approach genuinely learns distributed concept representations by generating concept contexts from *mentions* of those concepts in large encyclopedic text KBs such as *Wikipedia*. Utilizing such annotated KBs eliminates the need to manually annotate concept mentions and thus comes at no cost.

6.3.2 Concept Raw Context Model (CRX)

In this model, we jointly learn the embeddings of both words and concepts. First, all concept mentions are identified in the given corpus. Second, contexts are generated for both words and concepts from other surrounding words and other surrounding concepts as well. After generating all the contexts, we use the skip-gram model to jointly learn the embeddings of words and concepts. Formally, given a training corpus of V words w_1, w_2, \dots, w_V , we iterate over the corpus identifying words and

concept mentions and thus generating a sequence of T tokens t_1, t_2, \dots, t_T where $T < V$ (as multiword concepts will be counted as one token). Afterwards we train the a skip-gram model aiming to maximize:

$$\frac{1}{T} \sum_{i=1}^T \sum_{-s \leq j \leq s, j \neq 0} \log p(t_{i+j}|t_i) \quad (15)$$

where as in the conventional skip-gram model, s is the context window size. In this model, t_i is the target token which would be either a word or a concept mention, and t_{i+j} is a surrounding context word or concept mention.

This model is different from Yamada et al. [214]’s anchor context model in three aspects: 1) while generating target concept contexts, we utilize not only surrounding words but also other surrounding concepts, 2) our model aims to maximize $p(t_{i+j}|t_i)$ where t could be a word or a concept, while Yamada et al. [214] model maximizes $p(w_{i+j}|e_i)$ where e_i is the target concept/entity (see Yamada et al. [214] Eq. 6), and 3) in case t_i is a concept, our model captures all the contexts in which it appeared, while Yamada et al. [214] model generates for each entity one context of s previous and s next words. We hypothesize that considering both concepts and individual words in the optimization function generates more robust embeddings.

6.3.3 Concept-Concept Context Model (CCX)

Inspired by the distributional hypothesis [75], in this model, we hypothesize that: *”similar concepts tend to appear in similar conceptual contexts”*. In order to test this hypothesis, we propose learning concept embeddings by training a skip-gram model on contexts generated solely from concept mentions. As in the CRX model, we start by

identifying all concept mentions in the given corpus. Then, contexts of target concept are generated from surrounding concepts only. Formally, given a training corpus of V words w_1, w_2, \dots, w_V . We iterate over the corpus identifying concept mentions and thus generating a sequence of C concept tokens c_1, c_2, \dots, c_C where $C < V$. Afterwards we train the skip-gram model aiming to maximize:

$$\frac{1}{C} \sum_{i=1}^C \sum_{-s \leq j \leq s, j \neq 0} \log p(c_{i+j} | c_i) \quad (16)$$

where s is the context window size, c_i is the target concept, and c_{i+j} is a surrounding concept mention within s mentions.

This model is different from Li et al. [110] and Hu et al. [85] as they define the context of a target concept by all the other concepts which have an outgoing link from the concept’s corresponding article in *Wikipedia*. Clearly, some of these concepts might be irrelevant especially for very long articles which cite hundreds of other concepts. Our CCX model, alternatively, learns concept semantics from surrounding concepts and not only from those that are cited in its article. We also extend the context window beyond pairs of concepts allowing more influence to other nearby concepts.

6.3.4 CRX vs. CCX

One of the advantages of the CCX model over the CRX model is its *computational efficiency* during learning. On the other hand, the CCX model vocabulary is *limited to the corpus concepts* (all Wikipedia articles in our case), while the CRX model vocabulary is defined by *all unique concepts+words* in Wikipedia.

Table 20: Example three sentences along with sample contexts generated from CRX and CCX. Contexts are generated with a context window of length 3.

Sentence	CRX Contexts	CCX Contexts
<i>Larry Page</i> is the co-founder of <i>Google</i> which is headquartered in <i>Menlo Park CA</i>	< <i>Larry Page</i> , co-founder> <co-founder, <i>Google</i> > < <i>Google</i> , headquartered> <headquartered, <i>Menlo Park CA</i> >	< <i>Larry Page</i> , <i>Google</i> > < <i>Larry Page</i> , <i>Menlo Park CA</i> > < <i>Google</i> , <i>Menlo Park CA</i> >
<i>Bill Gates</i> is the co-founder of <i>Microsoft</i> which is headquartered in <i>Redmond WA</i>	< <i>Bill Gates</i> , co-founder> <co-founder, <i>Microsoft</i> > < <i>Microsoft</i> , headquartered> <headquartered, <i>Redmond WA</i> >	< <i>Bill Gates</i> , <i>Microsoft</i> > < <i>Bill Gates</i> , <i>Redmond WA</i> > < <i>Microsoft</i> , <i>Redmond WA</i> >
<i>Google</i> is headquartered in <i>Menlo Park CA</i> and was co-founded by <i>Larry Page</i>	< <i>Google</i> , headquartered> <headquartered, <i>Menlo Park CA</i> > < <i>Menlo Park CA</i> , co-founded> <co-founded, <i>Larry Page</i> >	< <i>Google</i> , <i>Menlo Park CA</i> > < <i>Google</i> , <i>Larry Page</i> > < <i>Menlo Park CA</i> , <i>Larry Page</i> >

Another distinct property of the CCX model is its emphasis on concept-concept *relatedness rather than similarity* (as we will detail more in the experiments section). The CCX model by looking only at surrounding concept mentions while learning, is able to generate contexts containing more diverse but related concepts. On the other hand, the CRX model which jointly learns the embeddings of words and concepts puts *more emphasis on similarity* by leveraging the full contextual information of words and concepts while learning.

To better illustrate this difference, consider a sample of the contexts generated from CRX and CCX in Table 20 using a sliding window of length 3. As we can notice, the CRX contexts of "*Google*" and "*Microsoft*" are somewhat similar containing words like "*headquartered*" and "*co-founder*". This causes the model to learn similar vectors for these two concepts. On the other hand, the CCX contexts of "*Google*" and "*Microsoft*" do not share any similarities⁵², rather we can see that "*Google*" has similar contexts to "*Larry Page*" as both has "*Menlo Park CA*" in their contexts, causing the model to learn similar embeddings for these two related concepts.

⁵²This is an illustrative example and doesn't imply the two concepts will have totally dissimilar vectors.

6.3.5 Training

We utilize a recent *Wikipedia* dump of August 2016³², which has about 7 million articles. We extract articles plain text discarding images and tables. We also discard "References" and "External links" sections (if any). We pruned both articles not under the main namespace and pruned all redirect pages as well. Eventually, our corpus contained about 5 million articles in total.

We preprocess each article replacing all its references to other *Wikipedia* articles with the their corresponding page IDs. In case any of the references is a title of a redirect page, we use the page ID of the original page to ensure that all concept mentions are normalized.

Following Mikolov et al. [139], we utilize negative sampling to approximate the softmax function by replacing every $\log p(w_O|w_I)$ term in the softmax function (Equation 14) with:

$$\log \sigma(\mathbf{v}_{w_O}^\top \mathbf{u}_{w_I}) + \sum_{s=1}^k \mathbb{E}_{w_s \sim P_n(w)} [\log \sigma(-\mathbf{v}_{w_s}^\top \mathbf{u}_{w_I})] \quad (17)$$

where k is the number of negative samples drawn for each word and $\sigma(x)$ is the sigmoid function ($\frac{1}{1+e^{-x}}$). In the case of the CRX model w_I and w_O would be replaced with t_i and t_{i+j} respectively. And in the case of the CCX model w_I and w_O would be replaced with c_i and c_{i+j} respectively.

For both the CRX & CCX models with use a context window of size 9 and a vector of 500 dimensions. We train the skip-gram model for 10 iterations using 12-core machine with 64GB of RAM. The CRX model took ~ 15 hours to train for a total of ~ 12.7 million tokens. The CCX model took ~ 1.5 hours to train for a total of ~ 4.5

million concepts.

6.3.6 Creating Continuous Bag-of-Concepts (CBoC)

As we mentioned in the related work section, the current mechanisms for BoC densification are inefficient as their complexity is at least quadratic with respect to the number of nonzero elements in the BoC vector. Here, we propose simple and efficient vector aggregation method to obtain fully continuous BoC vectors (CBoC) in linear time. Our mechanism works by performing a weighted average of the individual concept vectors in a given BoC. This operation has two advantages. First, it *scales linearly* with the number of nonzero dimensions in the BoC vector. Secondly, it produces a fully dense BoC vector of *fixed size* representing the semantics of the original concepts and *considering their weights*. Formally, given a sparse BoC vector $\mathbf{s} = \{(c_1, w_1), \dots, (c_{|\mathbf{s}|}, w_{|\mathbf{s}|})\}$, where w_i is weight of concept c_i . We can obtain the dense representation of \mathbf{s} as in equation 18:

$$\mathbf{s}_{dense} = \frac{\sum_{i=1}^{|\mathbf{s}|} w_i \cdot \mathbf{u}_{c_i}}{\sum_{i=1}^{|\mathbf{s}|} w_i} \quad (18)$$

where \mathbf{u}_{c_i} is the vector of concept c_i . Once we have this dense BoC vector, we can apply the cosine measure to compute the similarity between a pair of dense BoC vectors.

As we can notice, this weighted average is done *once* for any given BoC vector. Other mechanisms that rely on concept alignment [182], require *realignment* every time a pair of BoC vectors are compared. Our approach improves the *efficiency* especially in the context of dataless document classification with large number of

classes. Using our densification mechanism, we apply the weighted average for the BoC of each category and for each instance document once.

Interestingly, our densification mechanism allows us to densify the sparse BoC vector using only the *top few dimensions*. As we will show in the experiments (Section 6.5.3), we can get *near-best* results using these few dimensions compared to densifying with all the dimensions in the original sparse vector. This property reduces the cost of obtaining a BoC vector with a few hundred dimensions in the first place.

6.4 Text Conceptualization Applications

Concept-based representations have many applications in computational linguistics, information retrieval, and knowledge modeling. Such representations are able to capture the semantics of a given text by either identifying concept mentions in that text, transforming the text into a concept space, or both [209]. Thereafter, many cognitive tasks that require huge background and real-world knowledge are facilitated by leveraging the conceptual representations. We describe some of these tasks in this section, and provide empirical evaluation of our our concept embedding models on such tasks in the next section.

6.4.1 Concept/Entity Relatedness

Entity relatedness has been recently used to model *entity coherence* in many named entity linking and disambiguation systems [211, 142, 82, 30, 87, 85, 214]. In entity search, Hu et al. [85] utilized entity relatedness score to *rank* candidate entities based on their relatedness to the search query entities. Also, entity embeddings have proved more efficient and effective for measuring entity relatedness over traditional related-

ness measures which uses link analysis. Formally, given a entity pair (e_i, e_j) , their relatedness score is evaluated as $rel(e_i, e_j) = Sim(\mathbf{u}_{e_i}, \mathbf{u}_{e_j})$, where Sim is a similarity function (e.g., *cosine*), and \mathbf{u}_e is the embeddings of entity e .

6.4.2 Concept Learning

Concept learning is a cognitive process which involves classifying a given concept/entity to one or more candidate categories (e.g., *milk* as *beverage*, *dairy product*, *liquid...etc*). This process is also known as *concept categorization*⁵³ [110]. Automated concept learning gains its importance in many knowledge modeling tasks such as knowledge base *construction* (discovering new concepts), *completion* (inferring new relationships between concepts), and *curation* (removing noisy or assessing weak relationships). Similar to Li et al. [110], we assign a given concept to a target category using Rocchio classification [169], where the centroid of each category is set to the category’s corresponding embedding vector. Formally, given a set of n candidate concept categories $G = \{g_1, \dots, g_n\}$, a sample concept c , an embedding function f , and a similarity function Sim , then c is assigned to category g_* such that $g_* = arg \max_i Sim(f(g_i), f(c))$. Here, the embedding function f would always map the given concept to its vector.

6.4.3 Dataless Classification

Chang et al. [31] proposed dataless document classification as a *learning protocol* to perform text categorization without the need for labeled data to train a classifier. Given only label names and few descriptive keywords of each label, classification is

⁵³In this chapter, we use concept learning and concept categorization interchangeably

Algorithm 1: Classification + Bootstrapping

Input: $\mathbf{U} = \{(l_1, \mathbf{u}_{l_1}), \dots, (l_n, \mathbf{u}_{l_n})\}$: labels + embeddings
 $\mathbf{D} = \{(d_1, \mathbf{v}_{d_1}), \dots, (d_m, \mathbf{v}_{d_m})\}$: instances + embeddings
 N : number of bootstrap instances
Result: $\mathbf{L} = \{\dots, (d_i, l_j), \dots\}$: label assignment for each instance

```

1 repeat
2   candidates  $\leftarrow \{l_1 : \phi, \dots, l_n : \phi\}$ 
3   foreach  $(d, \mathbf{v}_d) \in \mathbf{D}$  do
4      $d_{max\_sim} = 0$ 
5      $d_{max\_label} = null$ 
6     foreach  $(l, \mathbf{u}_l) \in \mathbf{U}$  do
7        $sim_l = Sim(\mathbf{v}_d, \mathbf{u}_l)$ 
8       if  $sim_l > d_{max\_sim}$  then
9          $d_{max\_sim} = sim_l$ 
10         $d_{max\_label} = l$ 
11      end
12    end
13    add  $(d, d_{max\_sim})$  to candidates $_l$ 
14  end
15  foreach  $(l, candidates_l) \in candidates.items$  do
16    repeat
17       $score_{max} = 0$ 
18       $d_{max} = null$ 
19      foreach  $(d, score_d) \in candidates_l$  do
20        if  $score_d > score_{max}$  then
21           $score_{max} = score_d$ 
22           $d_{max} = d$  ▷ most similar instance so far
23        end
24      end
25      add  $(d_{max}, l)$  to  $\mathbf{L}$  ▷ assign class label
26       $\mathbf{u}_l \leftarrow \mathbf{u}_l + \mathbf{v}_d$  ▷ bootstrap label embedding
27      remove  $d$  from candidates $_l$ 
28      remove  $d$  from  $\mathbf{D}$ 
29    until  $N$  highest scored instances added
30  end
31 until  $\mathbf{D} = \phi$  ▷ no more instances to classify

```

performed *on the fly* by mapping each label into a BoC representation using ESA [56]. Likewise, each data instance is mapped into the same BoC semantic space and assigned to the most similar label using a proper similarity measure such as *cosine*. Formally, given a set of n labels $L = \{l_1, \dots, l_n\}$, a text document d , a BoC mapping model f , and a similarity function Sim . First we conceptualize each l_i and the document d by applying f on them, which will produce sparse BoC vectors s_{l_i} and s_d respectively. Then we densify the vectors as in equation 18 producing $s_{dense_{l_i}}$ and s_{dense_d} respectively. Finally d is assigned to label l_* such that $l_* = arg \max_i Sim(s_{dense_{l_i}}, s_{dense_d})$.

6.4.4 Bootstrapping

In the context of dataless classification, Chang et al. [31] and Song and Roth [181] used bootstrapping in order to improve the classification performance without the need for labeled data. The basic idea is to start from target labels as the initial training samples, train a classifier, and *iteratively* add to the training data those samples which the classifier is *most confident* until no more samples to be classified. The results of dataless classification with bootstrapping were competitive to supervised classification with many training examples.

We extend the use of bootstrapping to the concept learning task as well. In concept learning we start with the vectors of target category concepts as a prototype view upon which categorization decisions are made (e.g., $vec(bird)$, $vec(mammal)$...etc). We leverage bootstrapping by *iteratively updating* this prototype view with the vectors of concept instances we are most confident. For example, if "deer" is closest to "mammal" than any other instance in the dataset, then we update the definition of "mammal" by performing $vec(mammal) += vec(deer)$, and repeat the same operation for other categories as well. This way, we *adapt* the initial prototype view to better match the specifics of the given data. Although bootstrapping is a time consuming process, we argue that, using dense vectors for representing concepts makes bootstrapping more appealing. As updating the category vector with an instance vector could be performed through optimized vector arithmetic which is available in most modern machines. Algorithm 1 presents the pseudocode for performing dataless classification and concept categorization with bootstrapping. In our implementation, we bootstrap

the category vector with vectors of the most similar \mathbf{N} instances at a time. Another implementation option might be defining a threshold and bootstrapping using vectors of \mathbf{N} instances if their similarity score exceed that threshold. In the experiments, we set $\mathbf{N}=1$.

6.5 Experiments

6.5.1 Entity Semantic Relatedness

We evaluate the "goodness" of our concept embeddings on measuring entity semantic relatedness as an intrinsic evaluation.

6.5.1.1 Dataset

We use the KORE dataset created by Hoffart et al. [82]. It consists of 21 main entities from four domains: IT companies, Hollywood celebrities, video games, and television series. For each of these entities, 20 other candidate entities were selected and manually ranked based on their relatedness score based on human judgements. As in previous studies, we report the Spearman rank-order correlation (ρ) [222] which assesses how the automated ranking of candidate entities based on their relatedness score matches the ranking we obtain from human judgements.

6.5.1.2 Compared Systems

We compare our models with four previous methods:

1. **KORE** [82] which measure entity relatedness by firstly representing entities as sets of weighted keyphrases and then computing relatedness using different measures such as keyphrase vector cosine similarity and keyphrase overlap

Table 21: Evaluation of concept embeddings for measuring entity semantic relatedness using Spearman rank-order correlation (ρ). Overall, the CCX model gives the best results outperforming all other models. It comes 1st on 3 categories (bold), and 2nd on the other two (underlined).

Method	IT Companies	Celebrities	TV Series	Video Games	Chuck Norris	All
WLM	0.721	0.667	0.628	0.431	0.571	0.610
CombIC	0.644	0.690	0.643	0.532	0.558	0.624
ExRel	0.727	0.643	0.633	0.519	0.477	0.630
KORE	0.759	0.715	0.599	0.760	0.498	0.698
CRX	0.644	0.592	0.511	0.641	0.495	0.586
CCX	0.788	<u>0.694</u>	0.696	<u>0.708</u>	0.573	0.714

relatedness.

2. **WLM** introduced by Witten and Milne [211] who proposed a *Wikipedia* Link-based Measure (WLM) as a simple mechanism for modeling the semantic relatedness between *Wikipedia* entities. The authors utilized *Wikipedia* link structure under the assumption that related entities would have similar incoming links.
3. **Exclusivity-based Relatedness (ExRel)** introduced by Hulpus et al. [88] who proposed this measure under the assumption that not all instances of a given relation type should be equally weighted. Specifically, the authors hypothesized that the relatedness score between two concepts should be higher if each of them is related through the same relation type to fewer other concepts in the employed KB link graph.
4. **Combined Information Content (CombIC)** introduced by Schuhmacher and Ponzetto [175] who compute the relatedness score using a graph edit distance measure on the *DBpedia* KB.

Table 22: Top-3 rated entities from CRX & CCX models on sample entities from the 4 domains compared to the ground truth. We can notice high agreement between CCX model ranks and the ground truth ranks (in brackets). The CRX model top rated entities has lower ranks in ground truth causing relatively low correlation scores.

Entity	CRX	CCX	Ground Truth
Google	Yahoo! (9) Apple Inc. (12) Bing (search engine) (7)	<u>Larry Page</u> (1) <u>Sergey Brin</u> (2) Yahoo! (9)	Larry Page Sergey Brin Google Maps
Leonardo DiCaprio	Kate Winslet (4) Steven Spielberg (9) Tobey Maguire (7)	Tobey Maguire (7) Kate Winslet (4) <u>Titanic (1997 film)</u> (2)	Inception (film) Titanic (1997 film) Frank Abagnale
Mad Men	The Sopranos (15) <u>Matthew Weiner</u> (1) <u>Jon Hamm</u> (2)	<u>Matthew Weiner</u> (1) <u>Jon Hamm</u> (2) Todd London (4)	Matthew Weiner Jon Hamm Alan Taylor (director)
Guitar Hero (video game)	Frequency (video game) (10) Rock Band (video game) (6) <u>Harmonix Music Systems</u> (1)	<u>Harmonix Music Systems</u> (1) <u>WaveGroup Sound</u> (3) <u>RedOctane</u> (1)	Harmonix Music Systems RedOctane WaveGroup Sound

6.5.1.3 Results

Table 21 shows the Spearman (ρ) correlation scores of the CRX and CCX model compared to previous models. As we can notice the CCX model achieves the best overall performance on the five domains combined exceeding its successor KORE by 1.6%. The CRX model on the other hand came last on this task.

In order to better understand these results, we looked at rankings of individual entities from each domain to see how they compare to the ground truth. Table 22 shows the top-3 rated entities from each model on sample entities from the four domains. As we can notice, the ground truth assigns high rank to related rather than similar entities. For example, relatedness of "Google" to "Larry Page" is ranked 1st, while to "Yahoo!" is ranked 9th, and to "Apple Inc." is ranked 12th. As the CCX model emphasizes semantic relatedness over similarity, it has high overlap in the top-3 entities with the ground truth (underlined entities). On the other hand, the CRX model predictions are actually meaningful when it comes to functional

and topical similarity. As we can notice, it assigns high ranks of "Google" to other companies ("Yahoo!", "Apple Inc."), of "Leonardo DiCaprio" to other celebrities ("Tobey Maguire"), and "Mad Men" to other TV series ("The Sopranos"), and of "Guitar Hero" to other video games ("Frequency", "Rock Band"). However, all these highly ranked entities by CRX have relatively low rankings in the ground truth (given in brackets). This caused the correlation score to be much lower than what we obtained from the CCX model.

The results indicate that, the CCX model could be more appropriate in applications where relatedness and topical diversity are more desired than topical and functional coherence where the CRX model would be more appealing.

6.5.2 Concept Categorization

This task can be viewed as both intrinsic and extrinsic. It is intrinsic because a *good* embedding model would generate clusters of concepts belonging to the same category, and optimally place the category vector at the center of its instances vectors. On another hand, it is extrinsic as the embedding model could be used to generate a concept KB of is-a relationships with confidence scores, similar to *Probase* [212]. The model could even be used to curate and/or assert the facts in *Probase*.

6.5.2.1 Datasets

As in Li et al. [110], we utilize two benchmark datasets: 1) Battig test [11], which contains 83 single word concepts (e.g., *cat*, *tuna*, *spoon...etc*) belonging to 10 categories (e.g., *mammal*, *fish*, *kitchenware...etc*), and 2) DOTA which was created by Li et al. [110] from *Wikipedia* article titles (entities) and category names (categories).

DOTA contains 300 single-word concepts (DOTA-single) (e.g., *coffee*, *football*, *semantics...etc*), and (150) multiword concepts (DOTA-mult) (e.g., *masala chai*, *table tennis*, *noun phrase...etc*). Both belong to 15 categories (e.g., *beverage*, *sport*, *linguistics...etc*). Performance is measured in terms of the ability of the system to assign concept instances to their correct categories.

6.5.2.2 Compared Systems

We compare our model to various word, entity, and category embedding methods including:

1. **Word embeddings:** Collobert et al. [36] model (WE_{Senna}) trained on *Wikipedia*. Here vectors of multiword concepts are obtained by averaging their individual word vectors.
2. **MWEs embeddings:** Mikolov et al. [139] model ($WE_{Mikolov}$) trained on *Wikipedia*. This model jointly learns single and multiword embeddings where MWEs are identified using corpus statistics.
3. **Entity-category embeddings:** which include Bordes et al. [17] embedding model (TransE). This model utilizes relational data between entities in a KB as triplets in the form (entity,relation,entity) to generate representations of both entities and relationships. Li et al. [110] implemented three variants of this model (TransE₁, TransE₂, TransE₃) to generate representations for entities and categories jointly. Two other models introduced by Li et al. [110] are CE and HCE. CE generates embeddings for concepts and categories using category information of *Wikipedia* articles. HCE extends CE by incorporating *Wikipedia*'s

Table 23: Accuracy of concept categorization. The CRX model with bootstrapping gives the best results outperforming all other models.

Dataset/Instances	Battig	DOTA-single	DOTA-mult	DOTA-all
Method	(83)	(300)	(150)	(450)
WE _{Senna}	0.44	0.52	0.32	0.45
WE _{Mikolov}	0.74	0.72	0.67	0.72
TransE ₁	0.66	0.72	0.69	0.71
TransE ₂	0.75	0.80	0.77	0.79
TransE ₃	0.46	0.55	0.52	0.54
CE	0.79	0.89	0.85	0.88
HCE	0.87	0.93	0.91	0.92
CCX	0.72	0.90	0.80	0.87
+bootstrap	0.81	0.91	0.85	0.87
CRX	0.83	0.91	0.88	0.90
+bootstrap	0.89	0.98	0.95	0.97

category hierarchy while training the model to generate concept and category vectors.

6.5.2.3 Results

We report the accuracy scores of concept categorization⁵⁴ in Table 23. Accuracy is calculated by dividing the number of correctly classified concepts by the total number of concepts in the given dataset. Scores of all other methods are obtained from Li et al. [110]. As we can see in Table 23, the CRX model comes second after the HCE on all datasets. While the CCX model performance is much less than CRX. With bootstrapping, the CCX model performance improves on both datasets. CRX with bootstrapping outperforms all other models by significant percentages. These results show that learning concept embeddings from concept mentions is actually different from training the skip-gram model on phrases or multiword expressions. This

⁵⁴From a multi-class classification perspective, the accuracy scores would be equivalent to the clustering purity score as reported in Li et al. [110].

is clear from the significant performance gains we get from the CRX and CCX models compared to $WE_{Mikolov}$ which was trained using skip-gram on phrases. Additionally, the results demonstrate the efficacy of our models which simply learn concept embeddings from concept mentions in free-text corpus compared to the more complex models which require category or relational information such as TransE, CE, and HCE.

6.5.3 Dataless Classification

In this experiment, we evaluate the effectiveness of our concept embedding models on the dataless document classification task as an extrinsic evaluation. We demonstrate through empirical results the efficiency and effectiveness of our proposed BoC densification scheme which helps obtaining better classification results compared to the original sparse BoC representation.

6.5.3.1 Dataset

We use the 20-newsgroups dataset (20NG) [106] which is commonly used for benchmarking text classification algorithms. The dataset contains 20 categories each has ~ 1000 news posts. We obtained the BoC representations using ESA from Song and Roth [181] who utilized a Wikipedia index containing pages with 100+ words and 5+ outgoing links to create ESA mappings of 500 dimensions for both the categories and news posts of the 20NG. We designed two types of classification tasks: 1) fine-grained classification involving closely related classes such as *Hockey vs. Baseball*, *Autos vs. Motorcycles*, and *Guns vs. Mideast vs. Misc*, and 2) coarse-grained classification involving top-level categories such as *Sport vs. Politics* and *Sport vs. Religion*. The

Table 24: The 20NG dataset category mappings.

Top-level	Low-level
Sport	Hockey, Baseball, Autos, Motorcycles
Politics	Guns, Mideast, Misc
Religion	Christian, Atheism, Misc

Table 25: Evaluation results of dataless document classification of fine-grained classes measured in micro-averaged F1 along with # of dimensions (concepts) in the BoC at which corresponding performance is achieved.

Method	Hockey x Baseball		Autos x Motorcycles		Guns x Mideast x Misc	
ESA	94.60	@425	72.70	@325	70.00	@500
CCX (equal)	94.60	@20	-	-	70.33	@60
CRX (equal)	94.60	@60	73.10	@4	70.00	@7
WE_{max}	86.85	@65	76.15	@375	72.20	@300
WE_{hung}	95.20	@325	73.75	@300	71.70	@275
CCX (best)	95.10	@125	69.70	@7	72.47	@250
+bootstrap	95.90	@450	74.25	@12	77.43	@5
CRX (best)	95.65	@425	79.20	@14	73.40	@70
+bootstrap	95.90	@350	73.25	@12	77.03	@10

top-level categories are created by combining instances of the fine-grained categories which are shown in Table 24.

6.5.3.2 Compared Systems

We compare our models to three previous methods:

1. **ESA** which computes the cosine similarity between target labels and instance documents using the sparse BoC vectors.
2. **WE_{max}** & **WE_{hung}** which were proposed by Song and Roth [182] for BoC densification using embeddings obtained from Word2Vec. As the authors reported, we fix the minimum similarity threshold to 0.85. WE_{max} finds the best match for each concept, while WE_{hung} utilizes the Hungarian algorithm to find the

best concept-concept alignment on one-to-one basis. Both mechanisms have polynomial degree time complexity.

6.5.3.3 Results

Table 25 presents the results of fine-grained dataless classification measured in micro-averaged F1. As we can notice, ESA achieves its peak performance with a few hundred dimensions of the sparse BoC vector. Using our densification mechanism (equation 18), both the CRX & CCX models achieve equal performance to ESA at many fewer dimensions. Densification using the CRX model embeddings gives the best F1 scores on the three tasks. Interestingly, the CRX model improves the F1 score by $\sim 7\%$ using only 14 concepts on *Autos* vs. *Motorcycles*, and by $\sim 3\%$ using 70 concepts on *Guns* vs. *Mideast* vs. *Misc*. The CCX model, still performs better than ESA on 2 out of the 3 tasks. Both WE_{max} and WE_{hung} improve the performance over ESA but not as our CRX model.

When we applied bootstrapping, the performance of the CCX model improved slightly on *Hockey* vs. *Baseball*, but significantly ($\sim 5\%$) on the other two tasks achieving best performance on the third task with just 5 concepts. Bootstrapping with the CRX model has a similar effect to the CCX model except for *Autos* vs. *Motorcycles* where performance degraded significantly. To better understand this behavior, we analyzed the results as bootstrapping progresses at 14 concepts like CRX (best). We noticed that, at the very early iterations of Algorithm 1, many instances belonging to *Autos* were closer to *Motorcycles* with similarity scores between 0.90-0.95. And when using those instances to bootstrap *Motorcycles*, they caused *topic*

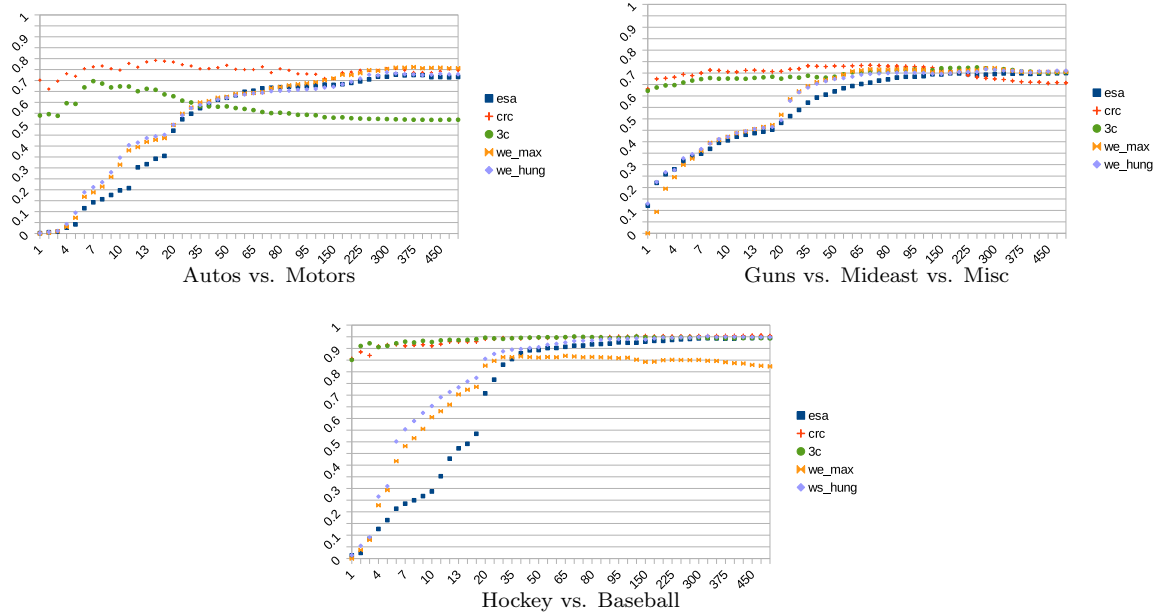


Figure 16: micro-averaged F1 scores of fine-grained classes when varying the # of concepts (dimensions) in the BoC from 1 to 500.

drift moving *Motorcycles*'s centroid toward *Autos*, and eventually causing relatively lower accuracy scores.

In order to better illustrate the robustness of our densification mechanism when varying the number of BoC dimensions, we measured F1 scores of each task as a function of the number of BoC dimensions used for densification. As we see in Figure 16, with *one* concept we can achieve high F1 scores compared to ESA which achieves *zero* or very low score. Moreover, *near-peak performance* is achievable with the top 50 or less dimensions. We can also notice that, as we increase the number of dimensions, both WE_{max} and WE_{hung} densification methods have the same undesired monotonic pattern like ESA. Actually, the imposed threshold by these methods does not allow for full dense representation of the BoC vector and therefore at low dimensions we still see low overall F1 score. Our proposed densification mechanism besides its low cost,

Table 26: Evaluation results of dataless document classification of coarse-grained classes measured in micro-averaged F1 along with # of dimensions (concepts) at which corresponding performance is achieved.

Method	Sport x Politics		Sport x Religion	
ESA	90.63	@425	94.39	@450
CCX (equal)	92.04	@2	95.11	@6
CRX (equal)	90.99	@2	94.81	@5
WE_{max}	91.89	@425	93.99	@425
WE_{hung}	90.89	@275	94.16	@450
CCX (best)	92.89	@4	95.86	@60
+bootstrap	93.20	@10	95.13	@225
CRX (best)	93.12	@13	95.91	@95
+bootstrap	92.96	@13	95.53	@70

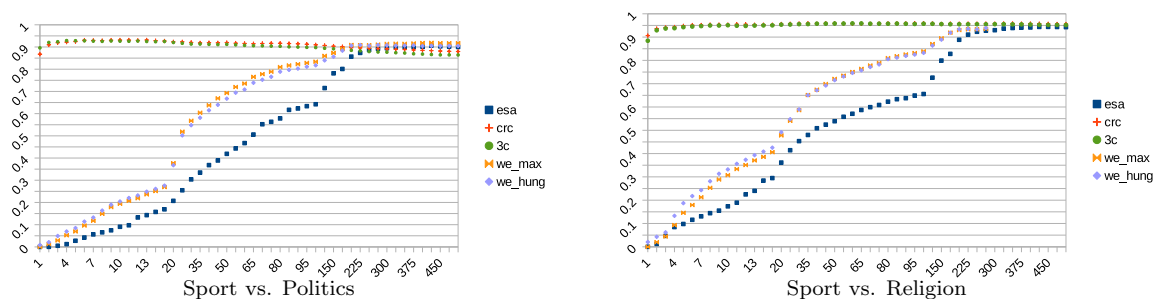


Figure 17: micro-averaged F1 scores of coarse-grained classes when varying the # of concepts (dimensions) in the BoC from 1 to 500.

produces fully densified representations allowing good similarities at low dimensions.

Results of coarse-grained classification are presented in Table 26. Classification at the top level is easier than the fine-grained level. Nevertheless, as with fine-grained classification, ESA still peaks with a few hundred dimensions of the sparse BoC vector. Both the CRX & CCX models achieve equal performance to ESA at very few dimensions (≤ 6). Densification using the CRX model embeddings still performs the best on both tasks. Interestingly, the CCX model gives very close F1 scores to the CRX model at less dimensions (@4 with *Sport vs. Politics*, and @60 with *Sport*

vs. *Religion*) indicating its competitive advantage when training computational cost is a decisive criteria. The CCX model, still performs better than ESA, WE_{max} , and WE_{hung} on both tasks.

Bootstrapping did not improve the results on this task significantly (if any). As we can notice in Table 26, the accuracy without bootstrapping is already high indicating that the initial prototype vector (centroid) of each class is representative enough of the instances to be classified.

Figure 17 shows F1 scores of coarse-grained classification when varying the # of BoC dimensions used for densification. The same pattern of achieving *near-peak performance at very few dimensions* recur with the CRX & CCX models. ESA using the sparse BoC vectors achieves low F1 up until few hundred dimensions are considered. Even with the costly WE_{max} and WE_{hung} densifications, performance sometimes decreases.

6.6 Discussion & Conclusion

In this chapter we proposed two models for learning neural embeddings of explicit concepts based on the skip-gram model. Explicit concepts are lexical expressions (single or multiwords) that denote an idea, event, or an object and typically have a set of properties associated with it. In the models presented here, our concept space is the set of all *Wikipedia* article titles. We proposed learning concept representations from concept mentions/references in *Wikipedia* making our models applicable to other open domain and domain specific free-text corpora by firstly wikifying⁵⁵ the text and

⁵⁵Wikification is the process of identifying mentions of concepts and entities in a given free-text and linking them to *Wikipedia*

Table 27: Top-5 related concepts from CRX & CCX models for sample target concepts.

Concept	Concept Raw Context (CRX)	Concept-Concept Context (CCX)
YouTube	Vevo Facebook SoundCloud Vimeo Viral video	Viral video Vimeo Vevo Video blog Dailymotion
Harvard University	Yale University Princeton University Brown University Columbia University Boston University	Harvard Kennedy School Cambridge, Massachusetts Harvard College Radcliffe College Harvard Society of Fellows
Black hole	Neutron star Accretion disk Primordial black hole Supermassive black hole Event horizon	Event horizon Neutron star Gravitational singularity Wormhole Hawking radiation
X-Men: Days of Future Past	X-Men: Apocalypse X-Men: First Class Deadpool (film) Avengers: Age of Ultron X-Men: The Last Stand	X-Men: Apocalypse The Wolverine (film) X-Men: First Class John Paesano William Stryker

then learning from concept mentions.

It is clear from the presented results that, the CRX model outperforms the CCX model on tasks that require topical coherence among the concepts vectors (e.g. concept categorization), while the CCX model is advantageous in tasks that require topical relatedness (e.g., measuring entity relatedness). To better show this difference qualitatively, we present a qualitative analysis of both models in Table 27 (target concepts are similar to those reported by Hu et al. [85]).

As we can notice, the CRX model tends to emphasize concept *topical and categorical similarity*, while the CCX model tends to more emphasize *concept relatedness*. For example, the top-5 concepts closest to "Harvard University" using CRX are all universities. While, the CCX model top-5 concepts include, besides educational in-

stitutions, location (*"Cambridge, Massachusetts"*) and an affiliated group (*"Harvard Society of Fellows"*). The same pattern can be noticed with the *"X-Men"* movie where we get similar genre movies with CRX. While we get related characters such as *"William Stryker"*⁵⁶ with CCX.

Based on these observations, we claim that the CCX model would be beneficial in situations where *diversity* is more desired than *topical coherence*. This claim is also supported by the results we obtained on the concept categorization and dataless densification tasks. On concept categorization, the performance gap between CRX and CCX was large with almost all datasets. On dataless classification, the performance gap was large with documents belonging topics with nuance differences (i.e., *Autos* vs. *Motorcycles*), but with other classes which have clear distinctions, the CCX performance was very competitive to CRX (e.g., *Hockey* vs *Baseball*).

In this chapter, we also proposed an *efficient* and *effective* mechanism for BoC densification which outperformed the previously proposed densification schemes on dataless document classification. Unlike these previous densification mechanisms, our method *scales linearly* with the number of the BoC dimensions. In addition, we demonstrated through the results how this efficient mechanism allows generating high quality dense BoC from few concepts alleviating the need of obtaining hundreds of concepts when generating the BoC in the first place.

Our learning method does not require training on a hierarchical concept category graph and is not tightly coupled to linked knowledge bases. Rather, we learn concept representations using mentions in free-text corpora with annotated concept mentions

⁵⁶https://en.wikipedia.org/wiki/William_Stryker

which even if not available could be obtained through state-of-the-art entity linking systems.

Finally, the work presented in this chapter serves two of our objectives: 1) it demonstrates utilizing textual knowledge bases to learn robust concept embeddings and hence increasing the *effectiveness* of the BoC representation to better capture semantic similarities between textual structures, and 2) it demonstrates utilizing the learned distributed concept vectors to increase the *efficiency* of the semantic representations in terms of space and computational complexities.

CHAPTER 7: LEVERAGING LARGE SCALE KNOWLEDGE BASES FOR LEARNING CONCEPT AND ENTITY REPRESENTATIONS

Text representation using neural word embeddings has proven efficacy in many natural language processing applications. As we showed in Chapter 6, we can adapt the traditional word embedding models to learn vectors of multiword expressions (concepts/entities⁵⁷) from their mentions in textual knowledge bases (e.g., *Wikipedia*). In this chapter, we propose a novel approach for learning concept vectors by integrating the knowledge from two large scale knowledge bases (*Wikipedia*, and *Probase*). We adapt the skip-gram model to seamlessly learn from the *Wikipedia* text and the *Probase* concept graph. We evaluate our concept embedding models intrinsically on two tasks: 1) analogical reasoning where we achieve a state-of-the-art performance of 91% on semantic analogies, 2) concept categorization on the two benchmark datasets used in Chapter 6, where we achieve a new state-of-the-art performance reaching categorization accuracy of 100% on one and 98% on the other. Additionally, we present a case study to extrinsically evaluate our model on unsupervised argument type identification for neural semantic parsing. We demonstrate the competitive accuracy of our unsupervised method and its ability to better generalize to out of vocabulary entity mentions compared to the tedious and error prone methods which depend on gazetteers and regular expressions.

⁵⁷In this chapter, we use the terms "concept" and "entity" interchangeably.

7.1 Introduction

As we mentioned in Chapter 4, vector-based semantic representation models are used to represent textual structures (words, phrases, and documents) as *multidimensional vectors*. Typically, These models utilize textual corpora and/or Knowledge Bases (KBs) in order to extract and model real-world knowledge. Once acquired, any given text structure is represented as a real-valued vector in the semantic space. The goal is thus to accurately place semantically similar structures close to each other in that semantic space, while placing dissimilar structures far apart.

Neural-based word embeddings stand out among these vector-based semantic representations as efficient and effective techniques which have succeeded in capturing both *syntactic and semantic regularities* using simple vector arithmetic [138, 139, 154]. Recently, a lot of research interest goes beyond word embeddings by focusing on learning distributed representations of concepts⁵⁸ and entities. Such models utilize text KBs (e.g., *Wikipedia*) or a triple-based KBs (e.g., *DBpedia* and *Freebase*) in order to learn entity vectors. Broadly speaking, existing methods can be divided into two categories. First, methods that learn embeddings of KB concepts only [85, 110, 167]. Second, methods that jointly learn embeddings of words and concepts in the same semantic space [26, 214, 24].

In this Chapter, we extend our concept embeddings models introduced in Chapter 6 introducing an effective approach for jointly learning word and concept vectors from two large scale KBs of different modalities; a text KB (*Wikipedia*) and a graph-based

⁵⁸concepts are lexical expressions (single or multiwords) that denote an idea, event, or an object and typically have a set of properties.

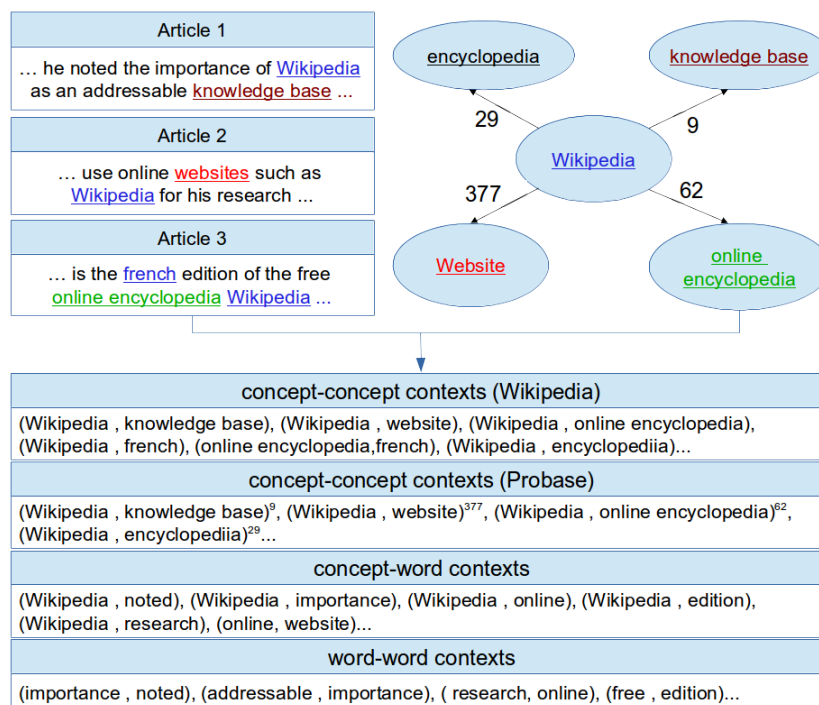


Figure 18: Integrating knowledge from *Wikipedia* text (left) and *Probase* concept graph (right). Local concept-concept, concept-word, and word-word contexts are generated from both KBs and used for training the skip-gram model.

concept KB (*Microsoft concept graph*⁵⁹ (aka *Probase*)). We adapt skip-gram, the popular local context window method [139], to integrate the knowledge from both KBs. As shown in Figure 18, three key properties differentiate our approach from existing methods. First, we generate word and concept contexts from their raw mentions in the *Wikipedia* text. This makes our model extensible to other text corpora with annotated concept mentions. Second, we model *Probase* as a *weighted undirected* knowledge graph exploiting the co-occurrence counts between pairs of concepts. This allows us to generate more concept-concept contexts during training, and subsequently learn better concept vectors for rare and infrequent concepts in *Wikipedia*. Third, to our knowledge, this work is the first to combine knowledge from two KBs of different

⁵⁹<https://concept.research.microsoft.com>

modalities (*Wikipedia* and *Probase*) into a unified representation.

We evaluate the generated concept vectors intrinsically on two tasks: 1) analogical reasoning where we achieve a state-of-the-art accuracy of 91% on semantic analogies, 2) concept categorization on the two benchmark datasets used in Chapter 6, where we achieve 100% accuracy on one dataset and 98% accuracy on the other. We also present a case study to analyze the impact of using our concept vectors for unsupervised argument type identification with semantic parsing as an end-to-end task. The results show competitive performance of our unsupervised method compared to the tedious and error prone argument type identification methods which depend on gazetteers and regular expressions. The analysis also shows superior generalization performance with utterances containing out of vocabulary (OOV) mentions.

We make our concept vectors and source code publicly available⁶⁰ for the research community for further experimentation and replication.

7.2 Learning Concept Embeddings

We learn continuous vectors of words and entities by building upon the skip-gram model [139] introduced in Section 6.3.1.

7.2.1 Learning from the Text

We use the exact learning approach proposed for the *Concept Raw Context* model (CRX) introduced in Section 6.3.2. We jointly learn the embeddings of both words and concepts using concept mentions. As described earlier, given a training corpus of V words w_1, w_2, \dots, w_V . We iterate over the corpus identifying words and concept

⁶⁰<https://sites.google.com/site/conceptembeddings/>

mentions and thus generating a sequence of T tokens t_1, t_2, \dots, t_T where $T < V$ (as multiword concepts will be counted as one token). Afterwards we train the a skip-gram model aiming to maximize:

$$\mathcal{L}_t = \frac{1}{T} \sum_{i=1}^T \sum_{-s \leq j \leq s, j \neq 0} \log p(t_{i+j}|t_i) \quad (19)$$

where s is the context window size. Here, t_i is the target token which would be either a word or a concept mention, and t_{i+j} is a surrounding context word or concept mention.

7.2.2 Learning from the Concept Graph

We employ *Microsoft concept graph (Probase)*, a large scale probabilistic KB of millions of concepts and their relationships (an is-a hierarchy). *Probase* was created by mining billions of Web pages and search logs of Microsoft’s Bing⁶¹ repository using syntactic patterns.

Probase is a different modality than *Wikipedia* because the knowledge is organized as a graph whose nodes are concepts and edges represent weighted is-a relationship between pairs of concepts. Formally, we model *Probase* as a 4-tuple graph $G = (C, E, \mathcal{T}_C, \mathcal{T}_E)$ such that:

- C is a set of vertices representing concepts.
- E is a set of edges (arcs) connecting pairs of concepts.
- \mathcal{T}_C is a finite set of tuples representing global statistics of each concept (i.e. its total occurrence count).

⁶¹<https://www.bing.com/>

- \mathcal{T}_E is a finite set of tuples representing co-statistics of each edge connecting pairs of concepts (i.e. their co-occurrence count).

Under this representation, location information about concepts is missing. Therefore the context of each concept can be defined by the set of its neighbors in the graph. Formally, the skip-gram optimization function would be maximizing:

$$\mathcal{L}_p = \frac{1}{|C|} \sum_{i=1}^{|C|} \sum_{(c_i, c_j) \in E} \log p(c_j | c_i) \quad (20)$$

thus *Probase* provides another source of conceptual knowledge to generate more concept-concept contexts and subsequently learn better concept representations.

7.2.3 Data and Model Training

We use the same *Wikipedia* dump of August 2016 used for training the CRX model. For *Probase*, we use its data repository⁶² which contains ~ 5 million unique concepts, ~ 12 million unique instances, and ~ 85 million is-a relationships. We follow a simple, exact string matching between *Wikipedia* article titles and *Probase* concept names, in order to align the concepts in both KBs and generate the final concepts set.

We call our model *Concept Multimodal Embeddings* (CME). During training, we jointly train our model to maximize $\mathcal{L} = \mathcal{L}_t + \mathcal{L}_p$ which as mentioned before is estimated using the softmax function (Equation 14) and implemented using negative sampling (Equation 17). For training, we use the same parameters used with CRX; a context window of size 9. We set the vector size to 500 dimensions and train the model for 10 iterations.

⁶²<https://concept.research.microsoft.com/Home/Download>

7.3 Evaluation

7.3.1 Analogical Reasoning

Mikolov et al. [140] introduced this intrinsic evaluation scheme to assess the capacity of the embedding model to learn a vector space with meaningful substructure. Typically, analogies take the form " *a to b is same as c to __?*" where *a*, *b*, and *c* are elements of the vocabulary *V*. Using vector arithmetic, this can be answered by identifying *d* such that $d = \arg \max Sim(vec(d), vec(b) - vec(a) + vec(c))$, $\forall d \in V \setminus \{a, b, c\}$ where *Sim* is a similarity function⁶³. A good performance on this task indicates the model's ability to learn semantic and syntactic patterns as linear relationships between vectors in the embedding space [154].

7.3.1.1 Dataset

We use the word analogies dataset [138]. The dataset contains 19,544 questions divided into semantic analogies (8,869), and syntactic analogies (10,675). The semantic analogies are questions about country capitals, state cities, country currencies...etc. For example, " *cairo to egypt is same as paris to france*". The syntactic analogies are questions about verb tenses, opposites, and adjective forms. For example, " *big to biggest is same as great to greatest*". In order to use the concept vectors, we first identify the corresponding entity of each analogy word and use its vector. If the word has no corresponding entity or corresponds to a disambiguation page under *Wikipedia* we use its word vector instead.

⁶³Cosine similarity or dot product if vectors are normalized.

Table 28: Evaluation results on the analogical reasoning task, given as percent accuracy. Our CME model gives the best result on semantic analogies and higher overall accuracy than all other models. Best performance (bold), second best (underlined).

Dataset/Questions	Semantic	Syntactic	All
Method	(8,869)	(10,675)	(19,544)
Word2Vec _{sg}	58	61	59.5
Word2Vec _{sg,b}	78.1	62.8	69.8
Glove	<u>80.8</u>	61.5	<u>70.3</u>
MPME	71.6	54.6	63.1
CME	91.4	<u>61.7</u>	75.2

7.3.1.2 Compared Systems

We compare our model to various word and entity embedding methods including:

1. **Word embeddings:** a) Word2Vec_{sg}, word embedding model trained on *Wikipedia* using skip-gram [138], b) Word2Vec_{sg,b}, a baseline model we created by training the skip-gram model on the same *Wikipedia* dump we used for our CME model, and c) GloVe, word embedding model trained on *Wikipedia* [154].
2. **Entity mention embeddings:** MPME is a recent model proposed by Cao et al. [26]. The model jointly learn embeddings of words and entity mentions by training the skip-gram on *Wikipedia* and utilizing anchor texts to generate multi-prototype entity mention embeddings.

7.3.1.3 Results

We report the accuracy scores of analogical reasoning in Table 28. As we see, our CME model outperforms all other models by significant percentages on the semantic analogies. The closest performing model (Glove) is $\sim 10\%$ less accurate. Performance on syntactic analogies is still very competitive to Word2Vec_{sg,b} and GloVe. Overall,

our model is $\sim 5\%$ better than the closest performing model.

7.3.1.4 Error Analysis

Local context window models like ours generally perform better on semantic analogies than syntactic ones. This indicates that syntactic regularities in most textual corpora are more difficult to capture than semantic regularities. A possible reason could be the more morphological variations of verbs and adjectives than nouns. Our model training is even more biased toward capturing semantic relationships between concepts by incorporating knowledge from *Probase* concept graph. This bias caused our model to produce more semantic predictions on the syntactic analogies than the $\text{Word2Vec}_{sg.b}$ baseline, returning a semantically related word to the answer. For instance, our model predicted "*fast*" rather than "*slows*" 9 times compared to 2 times by $\text{Word2Vec}_{sg.b}$. And "*large*" rather than "*smaller*" 14 times compared to 1 time by $\text{Word2Vec}_{sg.b}$. Another set of errors were predicting the correct word but with wrong ending especially "*ing*". For instance, "*implementing*" rather than "*implements*" 27 times compared to 19 time by $\text{Word2Vec}_{sg.b}$. We argue that, despite this bias, our CME model still produces very competitive performance compared to other models on syntactic analogies. And more importantly, emphasizing the semantic relatedness between concepts during training contributes to the significant accuracy gains on the semantic analogies.

7.3.2 Concept Categorization

We evaluate the CME model on the concept categorization task described in Sections 6.4.2 and 6.5.2. As described earlier, we assign a given concept to a target

Table 29: Evaluation results on the concept categorization task, given as percent accuracy. Our CME model with bootstrapping gives the best results outperforming all other models and baselines. Best performance (bold), second best (underlined).

Dataset/Instances	Battig	DOTA-single	DOTA-mult	DOTA-all
Method	(83)	(300)	(150)	(450)
WE _{Senna}	44	52	32	45
WE _{Mikolov}	74	72	67	72
TransE ₁	66	72	69	71
TransE ₂	75	80	77	79
TransE ₃	46	55	52	54
CE	79	89	85	88
HCE	87	93	<u>91</u>	<u>92</u>
WE _b	77	93	86	91
+bootstrap	88	<u>97</u>	86	90
CME	<u>94</u>	91	88	90
+bootstrap	100	99	95	98

category using Rocchio classification, where the centroid of each category is set to the category’s corresponding embedding vector. We also apply the bootstrapping algorithm (Section 6.4.4) to further boost the categorization accuracy without the need for labeled data. All experiments are performed on the same datasets described in Section 6.5.2.1. We also create a baseline model (WE_b) by training the skip-gram model on the same *Wikipedia* dump we used for our CME model.

7.3.2.1 Results

We report the accuracy scores of concept categorization in Table 29. Accuracy is calculated by dividing the number of correctly classified concepts by the total number of concepts in the given dataset. Scores of all other methods except WE_b are obtained from Li et al. [110]. As we can see in Table 29, our CME+bootstrap model outperforms all other models by significant percentages. And even achieves

100% accuracy on the Battig dataset. With single word concepts, CME achieves the best performance on Battig and competitive performance on DOTA-single. When it comes to multiword concepts, our CME model comes second after HCE.

7.3.2.2 Analysis

Is bootstrapping a magic bullet? A first look at the results of CME+bootstrap vs. CME might indicate that if bootstrapping is applied to HCE or WE_b which perform better than CME on some datasets, their performance would still be superior. However, the results of WE_b +bootstrap show that the margin of performance gains of bootstrapping is not necessarily proportional to the performance of the model without bootstrapping. For example, WE_b +bootstrap performs worse than CME_b +bootstrap on DOTA-single though WE_b was initially better than CME. This means that bootstrapping other better performing models such as HCE might not be as beneficial as it is to CME. The bottom line here that: the model should learn a semantic space with optimal substructures which clusters instances of the same category together and keep them far from instances of other categories. This is clearly the case with our CME model which ends up having (*near-*)*optimal* category vectors with bootstrapping.

7.3.3 Argument Type Identification: A Case Study

In this section, we present a case study to analyze the impact of using our concept vectors for unsupervised argument type identification with semantic parsing as an end-to-end task. In a nutshell, semantic parsing is concerned with mapping natural language utterances into executable logical forms [208]. The logical form is subsequently executed on a knowledge base to answer the user question. Table 30 shows

Table 30: Example utterances and their corresponding logical forms from the geography and flights domains. Left, utterances before and after argument type identification. Right, logical forms before and after argument type identification. City is mapped to c_i , Airport to ap_i , and River to r_i .

No	Utterance	Logical form
1	where is new orleans where is ci0	(lambda \$0 e (loc:t new_orleans_la:ci \$0)) (lambda \$0 e (loc:t ci0 \$0))
2	what states border the mississippi river how many states border ri0	(lambda \$0 e (and (state:t \$0) (next_to:t \$0 mississippi_river:r))) (count \$0 (and (state:t \$0) (next_to:t \$0 ri0)))
3	list flights from philadelphia to san francisco via dallas list flight from ci0 to ci1 via ci2	(lambda \$0 e (and (flight \$0) (from \$0 philadelphia:ci) (to \$0 san_francisco:ci) (stop \$0 dallas:ci))) (lambda \$0 e (and (flight \$0) (from \$0 ci0) (to \$0 ci1) (stop \$0 ci2)))
4	flights from jfk or la guardia to cleveland flight from ap0 or ap1 to ci0	(lambda \$0 e (and (flight \$0) (or (from \$0 jfk:ap) (from \$0 lga:ap)) (to \$0 cleveland:ci))) (lambda \$0 e (and (flight \$0) (or (from \$0 ap0) (from \$0 ap1)) (to \$0 ci0)))

some example utterances and their corresponding logical forms from the geography and flights domains.

7.3.3.1 Argument Identification

As we can notice from the examples in Table 30, user utterances usually contain mentions of entities of various types (e.g., *city*, *state*, and *airport* names). These mentions are typically parsed as arguments in the resulting logical form. Some of these mentions could be rare or even missing in the training data. As noted by Dong and Lapata [44], this problem reduces the model’s capacity to learn reliable parameters for such mentions.

One possible solution is to preprocess the training data replacing all entity mentions with their type names (e.g., *san francisco* to *city*, *california* to *state...etc*). This step allows the model to see more identical input-output patterns during training and thus better learn the parameters of such patterns. The model would also generalize better to out of vocabulary mentions because the same preprocessing could be done at test

time.

Dong and Lapata [44] proposed using gazetteers and regular expressions for argument identification. The authors also demonstrated increased accuracy when employing such approach. However, using regular expressions is *error prone*, as the same utterance could be paraphrased in many different ways. In addition, gazetteers usually have *low recall* and will not cover many surface forms of the same entity mention.

Our approach embraces argument type identification in a totally *unsupervised* fashion. The idea is to build upon the promising performance we achieved in concept categorization and apply the same scheme to map entity mentions to their corresponding type names. Our unsupervised argument type identification is a four step process: 1) we predefine target entity types and retrieve their corresponding vectors from our CME model, 2) we identify entity mentions in user utterances (e.g., *mississippi river*), 3) we lookup the mention vector in our CME model, and 4) we compute the similarity between the mention vector and each of the predefined target entity types and choose the most similar type if it exceeds a predefined threshold. This scheme is efficient and doesn't require any manually crafted rules or heuristics. The only needed parameter is the similarity threshold which we fix to 0.5 during experiments.

Note that, standard off-the-shelf entity recognition systems could help in identifying the entity mentions but not their type names. In domains like flights, we are interested in non standard types such as *airports* and *airlines*. It is also important to distinguish between *city*, *state*, and *country* mentions in the geography domain and not classifying all instances of these categories as the standard *location* type.

Table 31: Evaluation results of semantic parsing before and after argument type identification, given as percent accuracy. Using CME to identify argument types resulted in improved accuracy on both datasets.

Dataset	GEO	ATIS
w/o Identification	68.6	73.2
w/ Identification	77.1	83.7

7.3.3.2 Datasets

We analyze our unsupervised scheme on two datasets⁶⁴ : 1) GEO which contains a total of 880 utterances about U.S. geography [219]. The dataset is split into 680 training instances and 200 test instances. Here we target identifying five entity types: *city*, *state*, *river*, *mountain*, and *country*, and 2) ATIS which contains 5,410 utterances about flight bookings split into 4,480 training instances, 480 development instances, and 450 test instances. Here we target identifying six entity types: *city*, *state*, *airline*, *airport*, *day name*, and *month*.

7.3.3.3 Model & Training

We assess the performance of argument type identification by training Dong and Lapata [44] neural semantic parsing model⁶⁵. The model utilizes sequence-to-sequence learning with neural attention (see [44] for more details). We use the Seq2Seq variant of the model and do not perform any parameter tuning as our purpose is to analyze the performance before and after argument type identification not to get a state-of-the-art performance on these datasets.

⁶⁴We obtained the raw dataset files by contacting the authors of Dong and Lapata [44]

⁶⁵<https://github.com/donglixp/lang2logic>

7.3.3.4 Results

We report parsing accuracy in Table 31. Accuracy is defined as the proportion of the input utterances whose logical form is identical to the gold standard. As we can see, our argument type identification scheme resulted in significant accuracy improvements of $\sim 10\%$ on both datasets.

7.3.3.5 Error Analysis

Training the Seq2Seq semantic parsing model on preprocessed data is clearly beneficial and motivating as the results in Table 31 show. Without argument identification, the model is prone to the out of vocabulary problem. For example, on GEO we spotted 24 test instances with entities not mentioned in the training data (e.g., *new jersey*, *chattahoochee river*). The same on ATIS with 23 instances. Another source of errors was due to rare mentions. For example, "*portland*" appeared once in GEO training data.

Although our scheme demonstrated good ability to capture most entity mentions and map them to their correct type names. There was some subtle failure cases. For example, in "*what length is the mississippi*", our scheme mapped "*mississippi*" to the *state*, while it was mapped to the *river* in the gold standard logical form. Another example was mapping "*new york*" to the *city* in "*what is the density of the new york*", while it was mapped to the *state* in the gold standard.

Overall, the results show competitive performance of our unsupervised method compared to the tedious and error prone argument type identification methods. The analysis also shows superior generalization performance using unsupervised argument

identification with utterances containing out of vocabulary and rare mentions.

7.4 Conclusion & Discussion

Concepts are lexical expressions (single or multiwords) that denote an idea, event, or an object and typically have a set of properties associated with it. In this chapter we introduced a neural-based approach for learning embeddings of explicit concepts based on the skip-gram model. Our approach learns concept representations from mentions in free text corpora with annotated concept mentions which even if not available could be obtained through state-of-the-art entity linking systems. We also proposed an effective and seamless adaption to the skip-gram learning scheme in order to learn concept vectors from two large scale knowledge bases of different modalities (*Wikipedia*, and *Probase*).

We presented thorough evaluation of the learned concept embeddings intrinsically and extrinsically. Our performance on the analogical reasoning produced a new state-of-the-art performance of 91% on semantic analogies.

Empirical results on two datasets for performing concept categorization show superior performance of our approach over other word and entity embedding models.

We also presented a case study to analyze the feasibility of using the learned vectors for argument type identification with neural semantic parsing. The analysis shows significant performance gains using our unsupervised argument type identification scheme and better handling of out of vocabulary entity mentions.

To our knowledge, this work is the first to combine knowledge from both *Wikipedia* and *Probase* into a unified representation. Our concept space is all *Wikipedia* article

titles (~ 5 million). We use *Probase* as another source of conceptual knowledge to generate more concept-concept contexts and subsequently learn better concept vectors. In this spirit, we first filter *Probase* graph keeping only edges whose both vertices are *Wikipedia* concepts. Using string matching, ~ 1 million unique *Probase* concepts were mapped to *Wikipedia* articles. Note that, we still use the contexts generated from the 5 million *Wikipedia* concepts and add to them those contexts obtained from the filtered *Probase* graph. Out of the ~ 12.7 million vectors in our model, we have ~ 5 million concept vectors and ~ 7.7 million word vectors.

One important future improvement is to better match entities from both *Wikipedia* and *Probase*. For example, using string edits to increase recall or graph matching techniques to increase precision. Despite using the string matching, the performance of our method is superior compared to other methods utilizing *Wikipedia* only. It is expected that string matching might produce incorrect mappings. However, it is important to mention that our string matching exploits the redirect pages titles as well as the canonical titles of *Wikipedia* articles which increases the recall. For example, in *Probase*, *nyc*, *city of new york*, *new york city* are all matched with same *Wikipedia* article *New York City*.

Our initial qualitative analysis shows that it is common to match single-sense *Wikipedia* concepts (*ss-Wiki*) with multi-sense *Probase* concepts (*ms-Pro*) (e.g., *Tiger* and *Rose*). However, in many of these cases, the *ms-Pro* is dominated by the *ss-Wiki*. For example, *Wikipedia*'s page for *Tiger* describes the animal. In *Probase*, *Tiger* is-a *Animal* and *Tiger* is-a *Big cat* has more co-occurrences (917 & 315 respectively) compared to *Tiger* is-a *Dance* (1 co-occurrence). Same for *Rose* which is described

in *Wikipedia* as flowering plant. In *Probase*, *Rose* is-a *Flower* has (906) and *Rose* is-a *Plant* has (487) co-occurrences compared to *Rose* is-a *Garden* (10) and *Rose* is-a *Odor* (5) co-occurrences. We believe this would help generating more consistent contexts from *Wikipedia* and *Probase*. On the other hand, such multiple sense concepts in *Probase* could be used for tasks like sense disambiguation and multi-prototype embeddings along the lines of Camacho-Collados et al. [24], Iacobacci et al. [89], and Mancini et al. [132].

One important aspect of our CME model is its ability to better model the *long tail entities* with few mentions. Existing approaches that utilize *Wikipedia*'s link graph treat *Wikipedia* as unweighted directed KB graph. During training, a context is generated for entities e_1 and e_2 if e_1 has incoming/outgoing link from/to e_2 . This mechanism poorly models rare/infrequent *Wikipedia* concepts which have few incoming links (i.e. few mentions). We, alternatively, exploit *Probase* link structure modeling it as a weighted undirected KB graph. We also utilize the co-occurrence counts between pairs of concepts (see Figure 18). Therefore, we generate *more concept-concept* contexts resulting in better representations of the long-tail concepts. Consider for example *Nightstand* which has in *Wikipedia* 17 incoming links. In *Probase*, *Nightstand* is-a *Furniture*, is-a *Casegoods*, and is-a *Bedroom furniture* with co-occurrences 47, 47, and 32 respectively. This is a 100+ more contexts than we can generate from *Wikipedia*. Even for frequent *Wikipedia* concepts, our model by exploiting the co-occurrence counts will reinforce concept-concept relatedness from the many contexts obtained from *Probase*.

The work presented in this Chapter goes in harmony with the main theme of this

thesis, thus incorporating conceptual knowledge with the semantic representation in order to increase its effectiveness. As we demonstrated through empirical results, our concept-based embedding space gives superior performance over other pure word or pure entity embedding techniques. It also outperforms other embedding models which learn representations of words and entities from textual knowledge only.

CHAPTER 8: PATENT RETRIEVAL: A LITERATURE REVIEW

In this and the next Chapters, we focus on a very challenging text retrieval task; patent prior art search. We start with a literature review on patent retrieval in this Chapter. Then, we introduce a novel interactive framework for patent retrieval leveraging distributed representations in Chapter 9. We demonstrate through empirical results that superior patent retrieval performance can be achieved with interactive relevance feedback facilitated by our proposed concept-based representations.

8.1 Introduction

With the ever increasing number of filed patent applications every year, the need for effective and efficient systems for managing such tremendous amounts of data becomes inevitably important. Patent Retrieval (PR) is considered is the pillar of almost all patent analysis tasks. PR is a subfield of Information Retrieval (IR) which is concerned with developing techniques and methods that effectively and efficiently retrieve relevant patent documents in response to a given search request. In this Chapter we present a comprehensive review on PR methods and approaches. It is clear that, recent successes and maturity in IR applications such as Web search cannot be transferred directly to PR without deliberate domain adaptation and customization. Furthermore, state-of-the-art performance in automatic PR is still around average. These observations motivates the need for interactive search tools which provide cog-

nitive assistance to patent professionals with minimal effort. These tools must also be developed in hand with patent professionals considering their practices and expectations. We additionally touch on related tasks to PR such as patent valuation, litigation, licensing, and highlight potential opportunities and open directions for computational scientists in these domains.

Patents represent proxies for economic, technological, and even social activities. The Intellectual Property (IP) system motivates the disclosure of novel technologies and ideas by granting inventors exclusive monopoly rights on the economic value of their inventions. Patents, therefore, have a major impact on enterprises market value [157]. With the continuous rise in the number of filed patent applications every year, the need for effective and efficient systems for managing such tremendous amounts of data becomes inevitably important.

Typical patent analysis tasks include: 1) technology exploration in order to capture new and trendy technologies in a specific domain, and subsequently using them to create new innovative services, 2) technology landscape analysis in order to assess the density of patent filings of specific technology, and subsequently direct R&D activities accordingly, 3) competitive analysis and benchmarking in order to identify strengths and differences of corporate's own patent portfolio compared to other key players working on related technologies, 4) patent ranking and scoring in order to quantify the strength of the claims of an existing or a new patent, and 5) prior art search in order to retrieve patent documents and other scientific publications relevant to a new patent application. All those patent-related activities require tremendous level of domain expertise which, even if available, must be integrated with highly sophisticated

and intelligent analytics that provide cognitive and interactive assistance to the users.

Patent Retrieval (PR) is the pillar of almost all patent analysis tasks. PR is a subfield of Information Retrieval (IR) which is concerned with developing techniques and methods that effectively and efficiently retrieve relevant patent documents in response to a given search request. Although the field of IR has received huge advances from decades of research and development, research in PR is relatively newer and more challenging. On the one hand, patents are multi-page, multi-modal, multi-language, semi-structured, and metadata rich documents. On the other hand, patent queries can be a complete multi-page patent application. These unique features make traditional IR methods used for Web or ad hoc search inappropriate or at least of limited applicability in PR.

Moreover, patent data is multi-modal and heterogeneous. As indicated by Lupu et al. [118], analyzing such data is a challenging task for many reasons; patent documents are lengthy with highly complex and domain specific terminology. To establish their work novelty, inventors tend to use jargon and complex vocabulary to refer to the same concepts. They also use vague and abstract terms in order to broaden the scope of their patent protection making the problem of patent analysis linguistically challenging.

PR starts with a search request (query) which often represents a patent application under novelty examination. Therefore, several methods for query reformulation (QRE) have been proposed in order to select, remove, or expand terms in the original query for improved retrieval. QRE methods are either keyword-based, semantic-based, or interactive. Keyword-based methods work by searching for exact matches

between search query terms and the target corpus, and thus fail to retrieve relevant documents which use different vocabulary but have similar meaning to original query. In order to alleviate the vocabulary mismatch problem, semantic-based methods try to search by meaning through expanding queries and/or target corpus with similar or related terms and thus bridging the vocabulary gap. Because neither methods proved acceptable performance, few interactive methods were proposed to allow users to interactively control QRE with reasonable effort.

This review aims to provide researchers with an illustrative and critical overview of recent trends, challenges, and opportunities in PR. The rest of this Chapter is organized as follows. Section 8.2 presents some preliminaries and background about patents data. Section 8.3 provides an overview of evaluation tracks and data collections for PR benchmarking. An illustration of PR tasks is presented in Section 8.4. Section 8.5 presents a comprehensive review on PR methods and approaches. Section 8.6 lightly touches on related tasks such as patent quality assessment, litigation, and licensing. Finally, concluding remarks are presented in Section 8.7.

8.2 Preliminaries

8.2.1 Patent Documents and Kind Codes

Patent documents are mostly textual. They are highly structured with typical elements (sections) including *title*, *abstract*, *description* (aka *background of the invention*), and *claims*. The *description* section articulate in details the technical specification of the invention and its possible embodiments. The *claims* section is the most significant one as it describes the scope of protection sought by the inventor and hence

Table 32: Patent kind codes of major patent offices

Type	USPTO (US)	EPO (EP)	WIPO (WO)
A1	application	application w/ search report	
A2	republished application	application w/o search report	
A3	-	search report	
A4	-	supplementary search report	publication of amended claims
A9	modified application		
B1	granted patent w/o A1	granted patent (publication)	-
B2	granted patent w/ A1	amended B1	-

encodes the real value of the patent. Patent documents are lengthy with highly complex and domain specific terminology. They also contain multiple data types (e.g., text, images, flowcharts, formulae...etc) with a rich set of metadata and bibliographic information (e.g., *classification codes, citations, inventors, assignee, filing/publication dates, addresses, examiners...etc*).

Typically, each patent has a set of pertaining documents which published throughout its life-cycle. All documents are identified by an alphanumeric name with a common naming convention. Names start with two letters identifying the issuing patent office (e.g., US and EP), then the patent number as sequence of digits, and finally a suffix indicating the document's kind code. The kind code identifies the stage in the patent life-cycle at which the document is published. Table 32 shows a brief description of kind codes used at major patent offices including the US Patent and Trademark Office⁶⁶ (USPTO), the European Patent Office⁶⁷ (EPO), and the World Intellectual Property Organization⁶⁸(WIPO).

⁶⁶<http://www.uspto.gov/>

⁶⁷<http://www.epo.org/>

⁶⁸<http://www.wipo.int/portal/en/index.html>

8.2.2 Patent Classification

Patent offices organize patents by assigning classification codes to each of them based on the technical features of the invention. The patent classification system is a hierarchical one. Common classification systems include the International Patent Classification (IPC), the US Patent Classification (USPC), and the Cooperative Patent Classification (CPC).

8.2.3 Patent Families

A patent family is patent documents that refer to the same invention and are published by different patent offices around the world [159], usually in different languages depending on the issuing patent office. Patent families could be exploited to expand the prior art list of topic patents as they disclose the same invention.

8.3 Data & Evaluation Tracks

This section presents an overview of evaluation tracks organized for patent data analysis along with available data collections with focus on tasks pertaining to PR.

8.3.1 CLEF-IP Collections

The Conference and Labs of the Evaluation Forum⁶⁹ (CLEF) is an European series of workshops which started in 2001 to foster research in Cross Language Information Retrieval (CLIR). The Intellectual Property (IP) track (CLEF-IP) which ran between (2009-2013) was organized to: 1) foster research in patent data analysis, and 2) provide large and clean test collections of multi-language patent documents, specifically

⁶⁹<http://www.clef-initiative.eu/>

in the three main European languages (English, French, and German). Research labs have the opportunity to test their methods on multiple shared tasks such as PR, patent classification, image-based PR, image classification, flowchart recognition, and structure recognition [170, 156, 157, 158, 159].

The CLEF-IP data collection are patent documents extracted from the EPO data. It is provided through the Information Research Facility⁷⁰ (IRF) and hosted by Marec⁷¹. Patent documents are provided in XML format and have common Document Type Definition (DTD) schema. The collection was constructed according to the proposed methodology by Graf and Azzopardi [63] and is divided into two pools:

1. **The corpus pool:** documents selected from this pool are provided for participating labs as training or lookup instances depending on the task.
2. **The topics pool:** documents selected from this pool are called topics and they represent testing or evaluation instances depending on the task. For example, in prior art search, the topic might be a patent application document for which it is required to retrieve prior art.

The XML documents consist of the main textual sections such as *bibliographic data*, *abstract*, *description*, and *claims*. Each section is written in one or more languages (English, French, and/or German) and is denoted by a language code. At least the claims of granted patents (B1 documents) are written in the three languages because it is EPO requirement once a patent application is granted.

⁷⁰<http://www.ir-facility.org/>

⁷¹<http://www.ir-facility.org/prototypes/marec>

CLEF-IP 2009 Collection: this dataset was designed for the prior art search task [170]. The corpus pool contains documents published between (1985-2000) (~ 2 m documents pertaining to ~ 1 m unique patents). The topics pool contains documents published between (2001-2006) (~ 0.7 m documents pertaining to ~ 0.5 m individual patents). Topics are sets of documents from the topics pool with sizes ranging from 500 to 10,000 topics. Topics were assembled from granted patent documents including *abstract*, *description*, and *claims* sections. Citation information from the *bibliographic data* section was excluded.

A major pitfall in this dataset is its topics, which were chosen from granted patent documents (B1 documents). Initially, the creators of the dataset were motivated by having topics from granted patent documents which have claims in three languages. This was thought to provide a kind of parallel corpus suitable for CLIR. The problem of using such documents is simple, it contradicts the practice of IP search professionals who start with the patent application document not the granted one.

CLEF-IP 2010 Collection: this dataset was created for the prior art search and patent classification tasks [156]. The corpus pool of this dataset contains documents with publication date before 2002 (~ 2.6 m documents pertaining to ~ 1.9 m unique patents). The topics pool contains documents published between (2002-2009) (~ 0.8 m documents pertaining to ~ 0.6 m unique patents). Topics for the prior art task are two sets of documents from the topics pool; a small set of 500 topics and a larger set of 2000 topics. Unlike the CLEF-IP 2009 dataset, topics are assembled from patent application documents rather than granted patent documents.

CLEF-IP 2011 Collection: This dataset was created as a test collection for four

tasks: prior art search, patent classification, image-based prior art search, and image classification [157]. The topics and corpus pools were the same as in CLEF-IP 2010 dataset. For the prior art task, 3973 topics were provided as a separate archive of patent application documents.

CLEF-IP 2012 Collection: this dataset was created as a test collection for three tasks: passage retrieval starting from claims, chemical structure recognition, and flowchart recognition [158]. The topics and corpus pools were the same as in CLEF-IP 2010 dataset. The passage retrieval task is designed differently from previous CLEF-IP prior art search collections. The purpose for this tasks is to retrieve both documents and passages relevant to a set of claims. Topics for the passage retrieval task were extracted from patent applications published after 2001. Relevance judgments were the highly relevant citations only (i.e., marked X or Y) in the examiners' search reports (A4 documents) of chosen topic patents.

CLEF-IP 2013 Collection: this dataset was created as a test collection for two tasks: 1) passage retrieval from claims, and 2) structure recognition from patent images [159]. The topics and corpus pools were the same as in CLEF-IP 2010 dataset. Similar to CLEF-IP 2012, the CLM task is designed to retrieve both documents and passages relevant to a set of claims. Topics for the passage retrieval task were extracted from patent applications published after 2002. Overall, the topics set contained 148 topics extracted from 69 patent applications.

8.3.2 NTCIR Collections

The Japanese National Institute of Informatics Testbeds and Community for Information access Research project⁷² (NTCIR) started in 1997 to support research in IR and other areas, focusing on CLIR. NTCIR has been organizing a series of workshops providing test collections to researchers for evaluating their methodologies on multiple CLIR tasks [148]. Between NTCIR-3 and NTCIR-11 (2002-2013), there has been dedicated tasks for patent data analysis including patent retrieval [90], classification, mining, and translation.

NTCIR-3: the PR task in NTCIR-3 targeted the "technology survey" problem. The dataset for this task includes: 1) full text of Japanese patent applications between (1998-1999), 2) *abstract* of Japanese patent applications between (1995-1999) along with their respective English translations, and 3) 30 search topics where each topic includes a related newspaper article. The task is to retrieve patents relevant to news articles. Both cross-genre experiments in which patents were retrieved by a newspaper clip as well as ordinary ad hoc retrieval of patents by topics were conducted [90].

NTCIR-4: two PR tasks were organized in NTCIR-4 [52]: 1) patent map generation, and 2) invalidity search. The dataset for the PR tasks includes: 1) unexamined Japanese patent applications published between (1993-1997) along with English translations of the *abstract*, and 2) 34 search topics where each topic is a claim of a rejected patent application which was invalidated because of existing prior art. Relevance judgments were individual patents that can invalidate a topic claim by its own

⁷²<http://research.nii.ac.jp/ntcir>

or in conjunction with other patents. Relevant passages to the invalidated claim were also annotated and added to the relevance judgments.

NTCIR-5: two PR tasks were organized in NTCIR-5 [53]: 1) document retrieval (invalidity search), and 2) patent passage retrieval. The dataset for the invalidity search task includes: 1) unexamined Japanese patent applications published between (1993-2002) along with English translations of the *abstract*, and 2) 1200 search topics where each topic is a claim of an invalidated patent application. Relevance judgments were generated in a manner similar to the one used in NTCIR-4 invalidity search task.

NTCIR-6: two PR tasks were organized in NTCIR-6 [54]: 1) Japanese retrieval (invalidity search), and 2) English retrieval. The dataset for the Japanese retrieval task is the same one used in NTCIR-5 but more topics were used (1685 topics). The English retrieval task was focusing on finding all the citations cited by the applicant and the examiner. The dataset for this tasks includes: 1) granted patents from the USPTO between (1993-2000), and 2) 3221 search topics where each topic is a granted patent published between (2000-2001).

8.3.3 TREC-CHEM Collections

The TREC-CHEM track was organized to motivate large scale research on chemical datasets, especially chemical patent retrieval [116].

TREC-CHEM 2009: this collection was created as a test collection for two tasks [116]: 1) technology survey, and 2) prior art search. 18 topics were provided for the technology survey task where relevance judgments were obtained from experts and chemistry graduate students. For the prior art search, 1,000 patents were provided

as test topics where relevance judgments were collected from the citations of topic patents as well as their family members. The search corpus contains ~ 1.2 m chemical patents filed until 2007 at EPO, USPTO, and WIPO. It also contains 59K scientific articles.

TREC-CHEM 2010: this collection was created for the same two tasks as in TREC-CHEM 2009 [117]. 30 topics were provided for the technology survey task. The search corpus contains ~ 1.3 m chemical patents and 177K scientific articles. Relevance judgments were created the same way as in TREC-CHEM 2009.

TREC-CHEM 2011: this collection was created for the same two tasks as in previous TREC-CHEM tracks besides a new chemical image recognition task. The technology survey task topics were biomedical and pharmaceutical patents [119].

8.3.4 Other Sources

Other IP data sources are detailed by Schwartz and Sichelman [176]. These include full patent texts as well as bibliographic information from major patent offices such as the USPTO, EPO, and WIPO. Bibliographic information for patents published from 1976 to 2006 is provided through the National Bureau of Economic Research⁷³ (NBER) and subsequently cleaned and extended to include patents until 2013⁷⁴. Patent prosecution histories are available through the Patent Application Information Retrieval⁷⁵ (PAIR). Patent assignments, filings, classifications, and petition decisions are also provided through the USPTO bulk downloads previously hosted by

⁷³<https://sites.google.com/site/patentdatapoint/>

⁷⁴<http://rosencrantz.berkeley.edu/batchsql/>

⁷⁵<http://portal.uspto.gov/pair/PublicPair>

Google⁷⁶ and now by the USPTO⁷⁷.

8.4 Patent Retrieval Tasks

The goal of PR is to retrieve relevant patent documents to a given search request (query). This request can take different forms such as a sequence of keywords, a memo, or a complete text document (e.g. a patent application). The purpose of this task is manifold, for example:

- Retrieve related patents to a given patent application in order to gather related work, or invalidate one or more of its claims.
- Explore patent filing activity under specific technology.
- Explore the competitive landscape of a given company by looking at other companies filing patents similar to the given company patents.

Because of these multiple objectives, various PR tasks were proposed to fulfill each objective, and multiple datasets were provided depending on the given task.

Prior-art search is the main theme of the CLEF-IP and NTCIR tracks. The importance of this task stems from the requirement by all patent offices that filed patents must constitute novel, non-obvious, and non-abstract ideas. Therefore, an important activity through the patent life-cycle is to thoroughly ensure that no earlier published patent or material describing the prescribed ideas exist. The task can be defined as follows:

Problem: given a patent application X , retrieve all related documents to X .

⁷⁶<https://www.google.com/googlebooks/uspto-patents.html>

⁷⁷<https://www.uspto.gov/learning-and-resources/bulk-data-products>

Table 33: Scenarios of patent prior art search

Search Task	Who	When	Purpose	Output
Related Work	inventor/ prosecutor	pre-grant	all related work	applicant's disclosure
Patentability	prosecutor/ examiner	pre-grant/ examination	novelty breaking work	grant/modify/reject
Infringement	owner/ investor	post-grant	relevant claims/ infringing products	sue/license/clearance
Freedom to Operate	investor	post-grant	relevant claims/ related work	clearance
Invalidity	competitor/ defendant	post-grant	novelty breaking work	re-examine/ inter-parts review/ post-grant review
Technology Survey	technology analyst	pre/post-grant	all published patents	survey report

Prior art search is a total recall task, therefore it demonstrates several challenges. Search coverage is one of the main challenges, because it is required to cover all previously published material (patent or non-patent literature) in all forms (electronic or printed) which is infeasible. Another major challenge is the need to search through materials written in different languages. Last but not least, traditional IR methods perform poorly when confronted with the patent prior art search task. Mainly because the patent language is full of jargon and user defined terminology. Inventors intentionally tend to use different vocabulary to express same or similar ideas in order to establish the novelty of their work.

Prior art search is performed at different stages of the patent life-cycle, by different stakeholders, for various purposes, and for limited period of time. Understanding the real-life practices of patent professionals is critical to better satisfy their information need [94]. In other words, the search scenario depends on when it is done, by who,

and for what reason(s). Table 33 shows these various scenarios which are detailed below.

Related Work Search: during the pre-grant stage, inventors and prosecutors run related work search to retrieve all relevant work to the invention. Moreover, some patent offices request from inventors an applicant's disclosure document specifying all related publications when filing a new application.

Patentability Search: during the examination stage, patent examiners perform patentability search in order to ensure that the proposed ideas are novel, non-obvious, and non-abstract. The output of this task would be a search report with all retrieved relevant publications. In this report, each entry will have a special code indicating whether it is just a related publication, or novelty breaking one. Examiners would also specify which passages or figures in retrieved publications constitute relevancy. Depending on the search findings, the patent office might grant, reject, or ask the applicant to modify the patent application. Patentability search is also performed by patent prosecutors as a sanity check. Although this task should be of equal interest to prosecutors who file the patent application as it is to examiners, prosecutors often do not dig deep searching for relevant publications, and delegate finding relevant prior work to examiners in order to save costs.

Infringement Search: this task, also called product clearance search, aims to ensure whether an existing or a proposed product is infringing any published patent claim(s). Patent owners require that type search to find out if a third party has a product with features which are cited by one or more claims of their patents. If so, they might either sue or negotiate a license with that infringing party. The scope of

the search here would include all related publications to commercial products (e.g., product descriptions, vouchers...etc).

Investors and R&D managers, on the other hand, require that type of search to ensure newly proposed product(s) are not infringing a published patent claim(s) and investment in such products would be lucrative. The scope of search in this case would be limited to patent and copyrighted literature only. Deep understanding and correct interpretation of patent claims is imperative for building the correct correspondence between product features and claims in order to establish or dismiss infringement.

Freedom to Operate Search: this PR task extends beyond infringement search. Here, investors and R&D managers not only need to make sure that proposed products do not infringe an existing patent or copyrighted material, but also to ensure they have the freedom to file patents on these products without worrying about previously prior art that might invalidate such inventions. Another objective of freedom to operate search is to make better investment decisions and R&D plans according to existing prior art.

Invalidity Search: as patents guarantee monopoly rights to their owners on the economic value of granted inventions, companies and other parties usually monitor granted patents of their competitors or pertaining to their technology landscape to ensure competitive superiority. Therefore, invalidity search is performed to find published material that was missed by the patent office during patentability search. Invalidity search is also considered as the first line of defense when a party is confronted with patent infringement lawsuit. Again published material might include patent or non-patent literature such as books, news articles, academic periodicals...etc. After

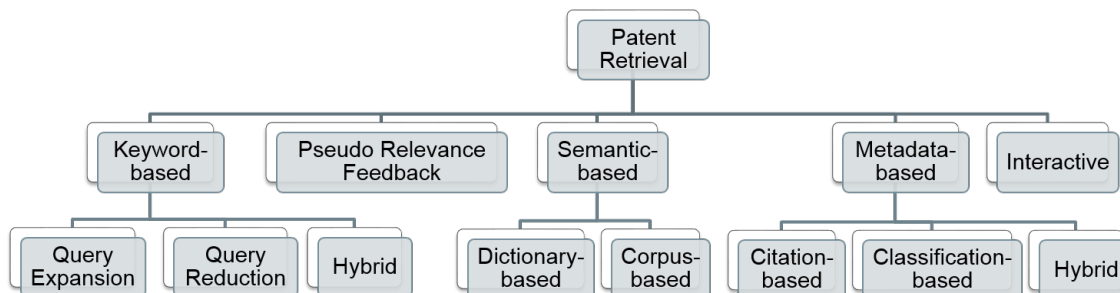


Figure 19: Taxonomy of patent retrieval methods.

finding such validity breaking material, a third party might file a post-grant (opposition) procedure depending on the patent office policies. For example, the USPTO provides procedures such as re-examination, Inter-partes review⁷⁸ (IPR), and Post-grant review⁷⁹ (PGR) in front of the Patent Trial and Appeal Board⁸⁰ (PTAB).

Technology Survey: another PR task where, in a typical scenario, business managers would request search professionals to prepare a survey of patent documents given a memorandum they prepared from some source (e.g., news article) [90]. This basic scenario is limited only to patent literature and it is assumed that patent documents are just a collection of technical papers.

8.5 Patent Retrieval Methods

In this section, we present a comprehensive review of PR methods and approaches. We start by presenting available test collections and evaluation metrics. Then, we

⁷⁸<http://www.uspto.gov/patents-application-process/appealing-patent-decisions/trials/inter-partes-review>

⁷⁹<http://www.uspto.gov/patents-application-process/appealing-patent-decisions/trials/post-grant-review>

⁸⁰<https://ptabtrials.uspto.gov>

provide a taxonomy of these approaches highlighting their characteristics and limitations.

As shown in Figure 19, PR methods can be categorized depending on which piece(s) of data from both the search queries and the search corpus are used for retrieving relevant documents. Keyword-based methods utilize only terms from search queries and look for exact matches in the target corpus. Pseudo Relevance Feedback methods utilize terms from the top ranked results of running the initial query to improve the set of relevant retrieved results. Semantic-based methods try to overcome the vocabulary mismatch problem between the search terms and related patents vocabulary by matching them based on their meanings. Metadata-based methods exploit the language independent non-textual metadata and bibliographic information in order to improve patent retrievability. Finally, interactive methods aim to better organize and present search results to the users. Moreover, through interaction, users are engaged in an iterative process of searching, reviewing, and refining hoping to retrieve as many relevant results as possible.

8.5.1 Test Collections & Evaluation Measures

As we highlighted in section 8.3, several datasets were created to support evaluating different PR techniques. In almost all of these datasets, relevant documents to search queries were collected from the citations of topic patent documents (e.g., CLEF-IP 2009/2010/2011 collections). Because these citations represent related prior work, they are appropriate only for the related work search task.

In other datasets such as CLEF-IP 2012/2013 collections, relevant documents were

collected from novelty breaking citations found in examiners' search reports, therefore, these datasets are appropriate for the patentability and invalidity search tasks. Though invalidity search requires non-patent literature as well.

Standard information retrieval as well as patent retrieval specific evaluation measures are generally used to evaluate patent retrieval systems including:

1. Precision (P) and Recall (R) at top- K ranks (e.g., $K=\{1, 5, 10, 50, 100, 1000\}$).
2. Mean Average Precision (MAP) [10] which generally favors early retrieval of relevant documents with less focus on recall.
3. Normalized Discounted Cumulative Gain (nDCG) [92] which favors not only early retrieval of relevant documents but also the respective ranking quality of these documents.
4. Patent Retrieval Evaluation Score (PRES) [122] which was proposed specifically for recall-oriented tasks such as PR. PRES focuses on the overall system recall as well as user's review effort which can be estimated from the rankings at which relevant documents are retrieved.

8.5.2 Query Reformulation (QRE)

The most widely used techniques for patent retrieval are the Query Reformulation (QRE) techniques. These methods aim at transforming the input query Q into \bar{Q} by means of reduction or expansion of Q terms in order to improve the retrievability of relevant documents. QRE can be performed through:

Table 34: Keyword-based patent retrieval methods.

Method	Description	Dataset	MAP	P	R	PRES
Verberne and D'hondt [200]	<ul style="list-style-type: none"> remove stopwords, and punctuation use claims as BOW 	clef-ip 2009	0.05	0.01	0.22	-
Magdy et al. [124]	<ul style="list-style-type: none"> remove stopwords, and frequent terms use different sections with manual weights perform IPC filtering use bigrams with $tf > 1$ 	clef-ip 2009	0.12	-	0.63	-
Mahdabi et al. [130]*	<ul style="list-style-type: none"> use query language models on different sections use queries of 100 terms perform IPC filtering 	clef-ip 2010	0.12	-	0.60	0.49
Wang and Lin [206]*	<ul style="list-style-type: none"> use linguistic-based concepts concept weighting using weighted tf-idf and mutual information 	clef-ip 2010	0.10	-	0.48	0.40
Konishi [99]	<ul style="list-style-type: none"> patterns to identify <i>claim</i> components terms patterns for explanation terms from <i>description</i> rank boosting based on IPC 	ntcir-5	0.20	-	-	-

* indicates scores @1000

- Query Reduction (QR):** where a representative subset of terms are selected from Q and used as \bar{Q} terms. Position-based methods are the most commonly used in this category where terms from specific parts or sections of the patent document are used, or given higher matching weight than others. Another example of query reduction is the IPC-based methods which utilize terms from IPC definitions as a lexicon or stop-words list for Q .
- Query Expansion (QE):** where representative terms other than the ones in Q are extracted and merged with Q to form \bar{Q} . Pseudo Relevance Feedback (PRF) methods are the most prominent in this category where terms from top ranked results of running Q are used to expand Q terms assuming these top results are relevant [25]. Other semantic-based query expansion methods work by expanding Q with terms of similar meanings such as synonyms or hyponyms.

- **Hybrid (query expansion & reduction):** where irrelevant terms are removed from Q and more relevant terms are appended to Q to form \bar{Q} . Most techniques used for query expansion are appropriate for query reduction as well, where only terms appearing in the expansion list are kept and all others are pruned.

8.5.2.1 Keyword-based Methods

This set of techniques retrieves relevant documents by looking for exact matches between search query term(s) and the target data. keyword search operates under the closed vocabulary assumption where vocabulary is derived solely from terms that appear in the target search data. Table 34 shows some keyword-based methods along with their performance results on benchmark datasets. Keyword-based techniques differ in: 1) which elements of the target data are indexed, 2) which query terms are selected/removed, 3) the relative weights of such terms, and 4) the match scoring function.

Query Reduction (QR): the rationale behind QR approaches is intuitive as patents are very long documents with several sections. Querying with the whole document would be impractical and inefficient. Some query reduction methods are position-based; they select relevant terms based on their position in the patent document [200, 42, 205, 124, 130]. For example, [200] used only terms from the *claims* section on the CLEF-IP 2009 collection. However, the results were moderate in terms in MAP compared to other runs on the same collection.

Magdy et al. [124] experimented using text from different sections of the topic

patent on the CLEF-IP 2009 collection. The authors used various combinations of sections including: 1) short sections such as *title*, *abstract*, first line of the *description*, first sentence of the *claims*, and 2) lengthy sections such as the *description* and the *claims*. The authors assigned different weights to each section manually. Their best scores were achieved using a combination of all short sections and post-filtering retrieved documents keeping only those that share the same IPC classification code with the topic patent. The main challenge with such approach is how to assign the respective weight of each section automatically. Moreover, IPC filtering wouldn't be possible when only partial patent application is available for prior art search.

Mahdabi et al. [130] proposed a position-based query reduction method which selects relevant query terms by building two query language models using various sections of the topic patent: 1) a variant of the weighted log-likelihood model [137], and 2) a model based on the parsimonious language model [79]. Their experiments showed that queries constructed from terms in the *description* section using weighted log-likelihood give better results than other sections which agrees with the previous results [213, 121, 18]. The main advantage of this approach is that, respective weights of query terms are derived automatically from the query model. However, some challenges still exist regarding tuning the model parameters such as the smoothing parameter which was set heuristically.

Query Expansion (QE): pattern-based QRE was proposed in many studies [150, 206, 99]. Wang and Lin [206] proposed patterns in the form of syntactic rules in order to extract query terms as weighted concepts. Konishi [99] proposed a pattern-based query expansion method for the patent invalidity search task on the NTCIR-5

Table 35: Pseudo relevance feedback patent retrieval methods.

Method	Description	Dataset	MAP	PRES
Bouadjenek et al. [18]	<ul style="list-style-type: none"> • use different methods of query expansion and reduction from the PRF set • use Rocchio, MMR, LM 	clef-ip 2010 clef-ip 2011	0.13 0.10	0.55 0.45
Magdy et al. [124]	<ul style="list-style-type: none"> • naive PRF • remove stop-words • use most frequent terms 	clef-ip 2009	0.05	-
Mahdabi and Crestani [126]	<ul style="list-style-type: none"> • build regression model using relevance score, RF similarities...etc • use the model to estimate the effectiveness of RF • use top 100 RF and maximize AP 	clef-ip 2010	0.16	0.56
Ganguly et al. [57]	<ul style="list-style-type: none"> • perform query segmentation • retain segments highly to be generated using RF LM 	clef-ip 2010	0.14	0.47
Golestan Far et al. [62]	<ul style="list-style-type: none"> • manually annotate one relevant RF result • add terms in the annotated result to the query 	clef-ip 2010	0.29 ⁸¹	-
Golestan Far et al. [62]	<ul style="list-style-type: none"> • assume relevant RF results are known • add terms more frequent in relevant than irrelevant RF to query 	clef-ip 2010	0.48 ⁸²	-

collection. Rather than using raw terms from topic patent’s *claims* which are often abstract, the author, using pattern matching, identifies other specific terms in the *description* and use them as expansion terms. First, components of the invention are extracted from the topic *claim* using handcrafted patterns. Secondly, explanation sentences describing components of the invention are extracted from the *description* using handcrafted patterns. Thirdly, terms from first and second steps are used as the new query. The results showed that this query expansion approach works better than using terms extracted from the *claims* section only. The main drawback of this method is its dependency on manually coded patterns to identify potential terms. Meanwhile, it demonstrates the potential of using entities and their relations as retrieval features motivating the need for deeper and more generic linguistic analysis of patent texts.

⁸³This is a semi-supervised performance

⁸⁴This is an Oracle performance

8.5.2.2 Pseudo Relevance Feedback (PRF)

These methods are one of the prominent techniques used for QRE. PRF starts with an initial run of the given query Q . Then, terms from top ranked results are used to select, remove, and/or expand terms in Q , assuming that these top results are relevant. PRF is thus advantageous as it works automatically without human intervention but might be computationally inefficient especially with long queries. Table 35 shows some PRF methods along with their performance results on benchmark datasets.

Despite their effectiveness and popularity, several challenges arise when it comes to PRF-based QRE [18] such as: 1) which part(s) of the patent application should be used as the initial query?; 2) which part(s) of the retrieved results should be used as the source of expansion and/or reduction?; 3) what is the best length of the expansion list in case of query expansion, or the best threshold for removing terms in case of reduction?; 4) which pseudo-relevant results are really relevant and how many of them should be used?; and 5) what is the best relevance scoring model for the search task (e.g., BM25 [168], the vector space model with tf-idf weighting...etc).

Bouadjenek et al. [18] provided a thorough evaluation on the CLEF-IP 2010/2011 collections to address some of the above challenges. The authors explored the scenario when only partial patent application is available for prior art search (e.g., *title*, *abstract*, *extended abstract*, or *description*). The authors tested different query expansion and reduction general methods such as Rocchio [174] and a variant of the Maximal Marginal Relevance (MMR) [27]. They also tested patent-specific methods utilizing synonym sets [123], language models [57], and IPC-based lexicon [131]. Af-

ter experimenting various sections as sources for the initial query terms as well as expansion/reduction sources, the results showed that, the *description* section among other sections is the best to use as the initial query in case of both query expansion and reduction. query reduction was not beneficial for the long *description* queries as it already contains good coverage of relevant terms. However, query reduction on *description* queries was useful as it removed many of the noisy terms. Generally, query reduction outperformed query expansion on *description* and *extended abstract* queries which indicates that, with long queries, query reduction is effective for better retrieval performance. The results also showed that generic query expansion methods such as Rocchio works generally better for query expansion than patent-specific query expansion methods. Finally, the results showed that BM25 scoring works better than the TF-IDF scoring on the long *description* queries for both query reduction and expansion, while TF-IDF works better than BM25 on short and medium-length *title* or *abstract* queries. Through this comprehensive experimental study, the authors did not evaluate the impact of using multiple sections in combination as sources for query expansion or reduction. More importantly, the study does not provide any insights about the respective values of number of expansion terms or term removal threshold and whether these values are somewhat deterministic or vary widely calling for interactive setting.

To address the problem of poor PRF results in patent retrieval compared to traditional information retrieval, Bashir and Rauber [13] proposed a novel approach for PRF-based query expansion which builds a model that learns to identify better PRF results based on their similarity with the query patent over specific terms. These terms

are learned by building a classification model that classifies whether a term would be useful for query expansion or not according to some proximity features between the original query terms and pseudo-relevant terms. The authors, through experiments on a subset of USPTO patents, showed the ability of this model to introduce more relevant query expansion terms and subsequently increasing the retrievability of individual patents. However, the authors did not evaluate this model on any of the available test collections. Moreover, extracting similarity features and computing similarities with PRF results during query execution is computationally expensive and time consuming.

Along the same efforts, Mahdabi and Crestani [126] proposed a framework for identifying effective PRF documents at runtime and then performing query expansion using terms from these relevant documents. The authors first proposed patent-specific features and then used them to build a regression model which calculates a relevancy score of each PRF document. Though results on the CLEF-IP 2010 collection were encouraging, several challenges still exist. For example, the computational complexity of calculating the regression model features at runtime. And PRF parameters tuning (e.g., number of PRF documents to use).

Ganguly et al. [57] proposed a PRF approach which utilizes a language model for query reduction of long queries composed of full patent applications. The authors argued that, naive application of PRF to expand query terms could add noisy terms causing query-topic drift. Moreover, naive removal of terms that has unit term frequency in the query could cause removal of useful terms and thus hurt retrieval effectiveness. Instead, the authors proposed a PRF-based query reduction technique

which generates language model similarity scores between query segments (sentences or n-grams) and top ranked results. Segments with top scores are kept and all others are removed. Results on the English subset of the CLEF-IP 2010 collection showed that the proposed approach outperforms the baselines. Parameter tuning is still the main downside of this technique. The performance of the proposed approach was unstable compared to the baselines with different parameter values. Specifically, the window size, the number of pseudo-relevant documents, and the fraction of terms to retain.

Golestan Far et al. [62] provided a study on hybrid QRE which aims to automatically approximate the optimal \bar{Q} by careful selection/expansion of relevant query terms. To motivate the efficacy of QRE on retrieval performance, the authors first designed an experiment where relevance judgments of a query patent Q were assumed to be known in advance. After running Q , using PRF on top- k documents, only terms that are more frequent in retrieved relevant documents (those from relevance judgments) than irrelevant documents are kept and used as \bar{Q} . Then, querying using \bar{Q} achieved a better performance than state-of-the-art on the English subset of CLEF-IP 2010 collection. To approximate \bar{Q} automatically, the authors proposed four different methods hoping to identify relevant vs. irrelevant terms in Q by: 1) removing terms with high document frequency in the top-100 retrieved documents, 2) removing infrequent terms in Q , 3) using frequent terms in relevant documents assuming the top-5 retrieved documents are relevant, and 4) performing query reduction on Q using IPC definitions as stop-words. All of the four methods failed to perform better than the keyword-based baseline. More interestingly, the authors demonstrated that, baseline

Table 36: Semantic-based patent retrieval methods.

Method	Description	Dataset	MAP	PRES
Magdy and Jones [123]	<ul style="list-style-type: none"> • use Wordnet synonyms and hyponyms for query expansion • slow processing time • no improvement 	clef-ip 2010	0.136 0.140*	0.484 0.486*
Tannebaum and Rauber [191] Tannebaum and Rauber [192] Tannebaum and Rauber [193] Tannebaum and Rauber [194] Tannebaum and Rauber [195] Tannebaum et al. [196]	<ul style="list-style-type: none"> • mine query logs for synonyms, co-occurring, and proximity terms • no improvement • use upon request 	clef-ip 2010	0.139 0.139*	0.512 0.512*
Magdy and Jones [123]	<ul style="list-style-type: none"> • using synonyms learned from parallel translations (EN, GE, and FR) • improve MAP only • use upon request 	clef-ip 2010	0.144 0.140*	0.485 0.486*

* indicates baseline performance

performance can be doubled if only one relevant document was manually provided by the user. This last observation motivates the need for interactive QRE as a simple and effective method for patent retrieval.

8.5.2.3 Semantic-based Methods

As we mentioned before, in PR queries can vary from few terms (e.g., survey memo) to thousands of terms (e.g., full patent application). Straightforward keyword-based PR proved to be ineffective simply because of the vocabulary mismatch between query terms and relevant patents content. [124] showed that, in the CLEF-IP 2009 collection, 12% of the relevant documents have no common words with the search topics. This motivates the need for novel approaches to bridge this vocabulary mismatch gap. Several semantic-based methods have been proposed in attempt to match queries with relevant documents based on their meanings rather than relying on keyword matches only. Table 36 shows some semantic-based methods along with their performance results on benchmark datasets.

Dictionary-based: semantic-based methods perform QRE by expanding the query to include other terms that have similar meanings to the original query terms. The first category of these methods are the dictionary-based techniques which use either generic [123], technical [114], or patent-specific dictionaries [192, 193, 195, 196, 204, 128] for QRE. Generic dictionaries could be existing lexical databases such as WordNet [49], while patent-specific dictionaries are lexical databases generated from patent-related data such as examiner's query logs. In either case, similar or related terms to the original query terms are retrieved from such dictionaries and used for query expansion.

Magdy and Jones [123] explored the use of WordNet for query expansion in PR on the CLEF-IP 2010 collection. Overall, adding synonyms and hyponyms for nouns and verbs in the original query increased the MAP score slightly, while decreased the PRES score significantly. Moreover, query execution time was increased considerably. The authors considered this a "negative" result. As the use of WordNet was proven to be effective in other retrieval tasks [203, 112], more experiments are needed to affirm the authors' conclusion. For example, investigating the impact of using synonyms only or hyponyms only, and expanding terms belonging to specific sections or ambiguous terms only.

Recently, more research was focused on utilizing domain-specific and technical dictionaries rather than WordNet. Examiners' query logs have been an important resource for building such technical thesauri. Tannebaum and Rauber [191, 192, 193, 194, 195] and Tannebaum et al. [196] introduced an analysis of the USPTO examiners' search query logs. Their analysis, though on a subset of query logs, revealed

interesting insights about patent examiners' search behavior which could be very useful for designing effective patent retrieval systems. For example, the authors noted that about examiners' behavior while searching for prior art: 1) the average query length is four terms, 2) search terms are mostly from the patent application under investigation, 3) expansion terms represent small percentage of query terms and mostly appear in the specific patent domain terminology, 4) the majority of query terms represents subject technical features that appears in the *claims* section, while very little percentage of them appears in the *description* section, 5) the majority of terms are nouns, followed by verbs, then adjectives, and 6) about half of the query operators used are "OR", followed by "AND", then proximity operators.

Tannebaum et al. built upon these insights and introduced methods to automatically identify synonyms/equivalents, co-occurring terms, and proximity relations for expanding query terms by mining examiners' search logs. As we can notice, learning expansion terms from query logs might be misleading because not all query sessions succeed to identify prior art. Additionally, deeper analysis of the query logs considering other metadata such as relevant hits count might be useful in this regard. On the other hand, it would be more useful if we can model the features of these terms, for example, based on their location, frequency, part-of-speech...etc. From effectiveness perspective, evaluating the generated lexical knowledge on the CLEF-IP 2010 collection did not record significant improvement [196]. Therefore, the authors recommended using it in an interactive mode rather than automatic mode to semi-automate query generation.

Corpus-based: the second category of semantic-based QRE is the corpus-based

methods. In these methods, textual corpora are analyzed to extract semantically related concepts to query terms which can be used for query expansion. Al-Shboul and Myaeng [4] proposed a Wikipedia-based query expansion method which works by first creating a summary of each Wikipedia article containing the main category, all titles under the main category, and other categories with in/out links to the main category. At query time, query terms and phrases are matched with page summaries, then, phrases from matching pages are scored and selected for query expansion under the assumption that they are semantically related. Experiments on the subset of USPTO patents in the NTCIR-6 collection showed increase in MAP over other query expansion techniques. However, the authors used IPC codes rather than citations as relevance judgments to topic queries which does not reflect the typical search practices, where it is needed to retrieve related patent documents not related classification codes.

Another corpus-based method was proposed by Magdy and Jones [123], where synonym sets were automatically generated from the CLEF-IP patent corpus. The authors utilized parallel translations of patent sections in different languages to build a word-to-word translation model and infer synonymy relation when a word in one language is translated to multiple words in another language. These multiple words under some probabilistic threshold could be considered synonyms. Overall results using this method were better than PRF and Wordnet based query expansion, but worse than the keyword-based baseline in Magdy and Jones [121]. The authors also showed that, the performance of this method on some topics was better than the baseline which indicates its potential. The issue they raised is how to more effec-

Table 37: Metadata-based patent retrieval methods.

Method	Description	Dataset	MAP	PRES
Fujii [51]	<ul style="list-style-type: none"> • use PageRank on patents citation graph • use patent popularity among top results with weighted voting 	ntcir-6	0.075 0.081 0.071*	-
Mahdabi and Crestani [127]	<ul style="list-style-type: none"> • build query specific citation graph from PRF results and their citations • weight nodes using PageRank • estimate query LM from the graph nodes considering their PageRank scores 	clef-ip 2011	0.105 0.099*	0.481 0.450*
Mahdabi and Crestani [129]	<ul style="list-style-type: none"> • using time-aware random walk on weighted citation graph 	clef-ip 2011	0.125 0.058*	0.536

* indicates baseline performance

tively apply query expansion by selecting "good" terms [25], or predicting query expansion performance beforehand [38, 120]. Such challenges can also be alleviated semi-automatically by developing intelligent and usable interactive query expansion frameworks which engage users in such decision. Finally, Krestel and Smyth [101] applied topic modeling of search hits in order to better rank retrieved patents. The results on a small collection of the USPTO patents showed improved MAP.

8.5.2.4 Metadata-based Methods

Patents are not only textual documents, they contain lots of non-textual metadata and bibliographic information as well (e.g., citations, tables, formulas, drawings, classification...etc). Combining metadata analysis with text-based PR has shown improvements in performance in the literature [51, 114, 115, 46, 127]. Metadata features are also language independent making them advantageous when used for CLIR. Table 37 shows some metadata-based methods along with their performance results on

benchmark datasets

Citation-based: The use of citation analysis for better retrieval is the most heavily reported technique of metadata-based methods. Naively incorporating citations from topic patent applications as prior art proved to be effective, eliminating the need for deeper citation analysis [125]. However, citation extraction from patent texts is challenging because there is no standard writing style for patent references. Lopez and Romary [115] developed a tool for citation mining which identifies, parses, normalizes, and consolidates patent citations. As citations might not be always available in all scenarios (e.g., related work search, technology survey...etc), more mature techniques are needed. Fujii [51] proposed using PageRank [20] and document popularity as an additional scoring to re-rank query top results returned using *claims*-based queries. The results of applying popularity scoring on the English subset of NTCIR-6 improved MAP and recall over the raw text-based scoring. Incorporating PageRank, though intuitive, poses many challenges especially because patent documents have references to non-patent literature which would produce incomplete citation graph. Mahdabi and Crestani [127] extended their query modeling technique in [130] by incorporating term distributions of the PRF results as well as their citations in calculating the query language model. The authors first construct a query-specific citation graph using PRF results and their citations and assign a score for each of them using PageRank. Then, a query model is estimated from term distributions of the documents in the citation graph constrained by their respective PageRank. Finally, query expansion is performed using the estimated query model. Experiments on the CLEF-IP 2011 collection showed improved recall performance with no change in precision, which

indicates the usefulness of using cited documents vocabulary for query expansion. Best improvements were achieved using the top 30 PRF documents, 2-levels citation graph, and 100 expansion terms. However, we can notice two main computational challenges using this technique in real-time setting: 1) computing the PageRank of the 2-level citations graph, and 2) estimating the query model from top PRF documents as well as documents in the citation graph.

Classification-based: these methods utilize classification information of the topic patent and the retrieved documents to improve the performance of patent retrieval[97, 73, 74, 32, 60]. The naive use of IPC classification is to filter retrieved documents to keep only ones that share the same IPC classification code at some level (e.g., same subclass) with the topic patent [124, 61]. More sophisticated use of classification information was introduced by Verma and Varma [201] who proposed a new representation of patent documents based on IPC classifications. The method utilizes IPC codes assigned to the corpus patents as well as codes of their citing documents to form an IPC class vector. First, the vector is initialized from patent's IPC code, then codes of citing patents are propagated over multiple iterations. The most similar patents are retrieved using cosine similarity between IPC class vectors and re-ranked using text-based search utilizing the top 20 tf-idf topic patent terms. Experiments on the CLEF-IP 2011 collection showed improved recall but low MAP scores. The instability of the patent classification system poses a real challenge when it comes to incorporating classification metadata into PR systems. Overtime, new classes are added to the classification hierarchy and existing classes are expanded. In order to do reliable search based on classification codes, these changes must be accounted for

periodically. Moreover, patents are assigned to multiple classification codes, however, almost all previous research considered only the primary class but not secondary classifications which might, if utilized, improve the retrieval performance.

Hybrid: these methods utilize various sources of metadata to improve PR performance. Mahdabi and Crestani [129] built upon previous work in [130] and [127] and proposed a query expansion method that utilizes time-aware random walk on a weighted patent citations network. Citation weights are derived from various metadata (e.g., classification codes, inventors, assignee...etc). Citations with higher weights are considered more influential when performing query expansion. Experiments on the CLEF-IP 2010/2011 collections show improved recall and MAP. Mahdabi and Crestani [128] proposed building a query-specific lexicon from IPC definition pages and using it for query expansion. Unfortunately, the lexicon would be helpful only if the query represents a complete patent document with IPC codes assigned to it which is not always the case especially at the early stages of the patent life-cycle.

8.5.2.5 Interactive Methods

Interactive patent retrieval is inevitable. As we can notice from the above review, effective fully automated retrieval of patent prior art is very challenging. Best methods perform around average in terms of PRES and much less in terms of MAP. Additionally, these methods require tuning a large number of parameters and thresholds whose optimal values differ according to the given query and the specific information need. For example, deciding which patent section to use?, which PRF results?, and which expansion terms and their respective weights?. The answers of these questions

are not deterministic and probably require multiple interaction cycles with the user in order to satisfy his/her information need.

Current interactive methods in patent retrieval are more focused on better organization, integration, and utilization of structured and textual patent data than on better retrieval performance. In other words, patent retrieval is addressed as a professional search problem rather than prior art search problem. Fafalios and Tzitzikas [47] presented a keyword-based interactive search framework to support patent search. The interaction elements are presented through post-analysis of search results in the form of facets based features like static metadata (e.g., IPC codes), textual clustering, named entity extraction, semantic enrichments, and others. The framework was applied on patent search [172] and evaluated using user study of twelve patent examiners [173]. Evaluation responses indicated overall acceptance of the framework in terms of usability, ease of use, efficiency, learnability. However, the authors did not report on the effectiveness or success of the system helping patent examiners to find prior art.

In Chapter 5, we proposed a visual interactive semantic framework for patent analysis which features semantic-based query expansion of search queries using Mined Semantic Analysis (MSA). As described, MSA builds an association knowledge graph using rule mining of concept rich textual corpora (e.g., Wikipedia). After mining the "See Also" link graph of Wikipedia, MSA could represent a topic query as a Bag-of-Concepts (BoC) derived from the association knowledge graph. This BoC could then be used to expand the original query terms (cf. Figure 12 which shows an example of the query expansion map of *Cognitive Analytics*, and cf. Figure 13 showing

concept map of 10 patents of *Bank of America* using the *abstract* section). Users can interact with the concept map by removing nodes and updating the search results. We demonstrated the applicability of this framework to support tasks such as prior art search, competitive intelligence, technology landscape analysis and exploration. In Chapter 9, we introduce a controlled study evaluating the performance of concept-based representations on benchmark patent retrieval collections.

Developing interactive methods for patent retrieval is also motivated by recent analysis which showed significant performance improvement if only one relevant document was manually provided by the user [62]. Performance gains using Technology Assisted Review (TAR) [64, 37] in domains like electronic discovery motivates investigating the applicability of machine learning TAR protocols in patent retrieval.

Technology assisted review, like patent retrieval, is a total-recall task where it is required to find all relevant documents to the search request with reasonable effort (time and cost). It is thus a human-in-the-loop process where a human expert manually annotates a subset of the documents as relevant or irrelevant. The underlying algorithm subsequently builds a ranking model by training on such annotations and uses this model to promote more relevant results and demote irrelevant ones as more documents are searched and annotated. This process stops when enough results are obtained. Typically, these algorithms utilize techniques such as continuous active learning combined with Boolean search in order to develop and adapt the ranking model [64].

Several questions still need to be addressed when it comes to investigating technology assisted review protocols applicability to patent retrieval, as these protocols were

only evaluated in ad hoc search scenarios. The complexity of patents terminology and availability of multiple sources of metadata would, likely, demonstrate many opportunities for adaptation and modifications to the current technology assisted review protocols.

8.6 Related Topics

Despite intense interest within the research community in patent retrieval, the patent industry has many other challenges and open problems which are of high interest and value to various stakeholders, such as economists, R&D managers, and legal professionals, to name a few. In this section, we try to lightly touch on these tasks and highlight some challenges and possible future directions.

8.6.1 Patent Quality Assessment

Assessing the technical quality and importance of inventions is very important to patent owners because it allows them to:

- better utilize their IP management costs by automated recommendation of patent maintenance decisions.
- better determine the novelty and originality of their patents.
- maximize licensing revenues by automatic estimation of the patent value.

Because there is no ground truth for quality measurements, performance evaluation of quality assessment techniques is usually based on indicators such as correlation with patent forward citations, maintenance status history, court rulings (if any), and/or patent reexamination history (if any). Some early work scored patents using their

metadata such as citations count, maintenance history, global prosecution efforts [107], and even manually by patent attorneys. Automated patent quality assessment has gained more traction in recent years though.

Citation analysis has been and still a main technique for patent valuation [198, 72, 68, 40, 207]. Wang et al. [207] proposed a probabilistic mixture approach to predict whether a topic patent will be renewed at different renewal periods. The method first divides the citations into two groups; technological and legal. From each group, different features reflecting the technological richness, technological influence, legal patent scope, and legal blocking power of each patent are combined. The authors subsequently build a binary classifier using these probabilistic features. Evaluation is performed by comparing the model's predictions against the renewal decisions of a collection of patents. While proved effectiveness, estimating patent value as a binary outcome might not be practical especially if a patent owner needs to prioritize his maintenance decisions of multiple patents.

Quality assessment based on the lexical features of the patent text was also explored in the literature [93, 113, 78]. Liu et al. [113] proposed a graphical model to estimate patent quality as a latent variable. The model utilized lexical features extracted from the patent text such as *claims* n-grams age and popularity, lexical alignment between the *claims* and the *description*, number of dependent and independent claims, number of reported classes when filing the patent, and other features. The authors also incorporated measurements such as forward citations count, court decisions, and reexamination records. It is clear that court decisions are only available for small number of patents which might not allow building a robust model.

Jin et al. [93] modeled the patent maintenance decision as recommendation problem where patents were represented as multimodal heterogeneous information network. The model utilized several metadata features, lexical features such as unique words and lengths of different sections, as well as inventor and assignee profile features. Experimental results showed high prediction accuracy on a large number of USPTO patents.

Hu et al. [84] proposed a time-based topic model which ranks patents novelty and influence based on whether the dominant topics in patent's prior art (for novelty) or forward art (for influence) are still active topics. The authors also proposed using time decay function to address the problem of old patents having less prior art and more forward art than newer patents and vice versa. Results showed high correlation between assigned ranks and forward citations count.

Hido et al. [78] proposed a scoring model which assigned a patentability score to each patent and thus can be utilized to determine whether it will be granted. First, the authors extracted textual features such as word frequency, word age, and syntactic complexity (e.g., number of sentences). Then, they trained a classifier using previous patent office decisions as ground truth. Though results showed the model effectiveness, the utilized syntactic complexity features are all extracted from the topic patent and thus could be good predictors for the writing quality not patentability potential.

The correlation between patent *claims* novelty and patent value using lexical analysis of patent text has been analyzed in previous studies [33, 76]. Hasan et al. [76] proposed an IR-based ranking tool which analyses patent *claims* for originality. The

technique first extracts key terms and phrases from the *claims* text using syntactic patterns and then looks for usage patterns backward to determine their novelty, and forward to determine their influence. The method considers usage patterns only through user defined time window. It is also keyword-based and hence will fail to capture key phrases that are semantically similar and subsequently might give inaccurate scores.

Along the efforts of using patent legal data for quality assessment, Mann and Underweiser [133] utilized prosecution histories, court decisions, and patent textual features to analyze patent quality. The analysis suggested that patent examination records would be very helpful in better discriminating high from low quality patents and possibly improve the examination process as a whole.

8.6.2 Patent Litigation

Litigation in general, and patent litigation specifically have been and still a topic of interest to legal professionals. With the increased amounts of digitized data available and the need for technology support in analyzing and mining these huge datasets, litigation became of more interest to computational science researchers. Patent litigation can take many forms, the most common is patent infringement litigation where a patent owner (plaintiff) accuses another party (defendant) of using his/her invention without license or permission. Because litigation is very expensive, the most common defensive action for the defendant is to establish invalidity of the plaintiff invention by issuing a post grant proceeding such as post-grant review or inter-parts review. Now the problem becomes a patent retrieval task, i.e. invalidity search, where one

of the aforementioned methods can be utilized with wider scope to cover not only patent literature but also other published material.

The task of automatically establishing patent infringement is not addressed in literature. Such task requires extensive human expertise and reasoning to build correspondences between product features and patent claims. On the other hand, statistical and visual analytics of previous court decisions have shown some degree of success in helping lawyers to better understand possible outcomes and better plan on defense strategies [71, 8, 149].

For example, Allison et al. [9] provided a statistical study on patent cases filed from 2008 to 2009 and decisions made between (2009-2013). The study showed that, there is a strong correlation between court decision, and patent-specific, litigation-specific, and industry-specific variables such as industry and technology type, inventors foreign status, number of claims, number of forward and backward citations, and number of defendants sued.

Rajshekhar et al. [162] studied the potential of concept-based semantic search in patent litigation. The authors designed an experiment in order to retrieve invalidating patents to a given litigated patent using a subset of PTAB's final decisions as ground truth and a search corpus of ~ 7 m USPTO patents. The authors, based on the experimental results and through interviews with patent practitioners, concluded that, a one-size-fits-all semantic search approach is incapable of capturing the highly nuanced relevance judgments made in the domain of patent litigation. Rather, the search workflow should be modeled as a multistage information seeking process, where users are presented with interactive elements to control the search space, and their

feedback is incorporated iteratively in the relevance ranking of retrieved results for enhanced performance.

Finally, There is much to be done in building predictive models for patent litigation given the availability of prior case datasets that were not available few years ago (e.g., prosecution histories, court decisions, and PTAB decisions).

8.6.3 Technology Licensing

Patents represent one of the most valuable assets in today's enterprises which, if leveraged effectively, guarantee not only competitive superiority, but also huge licensing revenues [33]. The technology licensing task is three sided. First, patent owners would be interested in finding potential licensees with reasonable effort. Second, licensees would like to find relevant inventions to their businesses. Third, owners and businesses would be interested in gauging the strategic and protection values of a patent in order to support their pricing and offering decisions.

While there is no much research focusing on the task of automatically recommending potential licensees. The task of recommending patents to be licensed was relatively more considered. Chen et al. [33] proposed a platform called SIMPLE which is used at IBM to identify target patents for licensing. Given a set of topic patents, SIMPLE uses nearest neighbor similarity to find other patents that are most similar to the given topic set. Then, all the patents are grouped and proposed as one licensing package to interested party. The platform was extended in Spangler et al. [186] to allow retrieving target patents using free text search. We can notice that current trends for identifying potential patents for licensing model the problem as a PR task.

More elaboration on the SIMPLE platform was introduced by Spangler et al. [187] using interactive visualization. First, portfolios of two companies are contrasted to find content overlap between both of them using proximal search. Then, the closest patents to the overlap area are recommended as candidates for licensing.

8.7 Concluding Remarks

In this Chapter we presented a comprehensive review of patent retrieval methods and approaches. It is clear that, the well-performing information retrieval techniques in areas like Web search cannot be utilized directly in PR without deliberate domain adaptation and customization. Furthermore, state-of-the-art performance in automatic patent retrieval is still low (<0.2 MAP). Several proposed techniques for query expansion, query reduction and pseudo relevance feedback require tuning of various parameters. Search professional practices suggest that effective prior art search requires multiple iterations of searching, reviewing, and refining. On the other hand, examiners' query formulation practices (few keywords and Boolean search) are different from those of automatic methods (many keywords and free-text search). These observations motivate the need for interactive search tools which provide cognitive assistance to search professionals with minimal effort. These tools must also be developed in hand with patent professionals considering their practices and expectations.

Unexplored patent-related data sources might be an opportunity for breakthrough improvements over the current modest state-of-the-art in patent retrieval. For example, utilizing reexamination records, PTAB decisions, differences between the patent application and the granted version, examiner/applicant correspondences, and pros-

ecution histories. All these resources are not yet fully explored in the literature of patent retrieval.

Related tasks such as patent quality assessment, litigation, and licensing are of less focus among computational scientists. However, they provide wide opportunities for future exploration from computational and modeling perspectives. These tasks require interdisciplinary and cooperative efforts from both legal professionals and the computer science research community.

CHAPTER 9: TOWARD AN INTERACTIVE PATENT RETRIEVAL FRAMEWORK BASED ON DISTRIBUTED REPRESENTATIONS

In Chapter 2 we addressed increasing the efficiency of technical text representation through unsupervised concept-based dimensionality reduction. We also introduced in Chapter 6 and Chapter 7 three neural-embedding models for generating dense concept-based representations. These models are more efficient than the sparse representations (e.g., bag-of-words), and also capture semantic and syntactic regularities which are desired requirements for almost all text analysis and retrieval tasks. In this chapter we demonstrate the effectiveness and usability of these representations for technical text retrieval. We present a novel interactive framework for patent retrieval leveraging distributed representations of concepts and entities extracted from the patents text. We propose a simple and practical interactive relevance feedback mechanism where the user is asked to annotate relevant/irrelevant results from the top n hits. We then utilize this feedback for query reformulation and term weighting where weights are assigned based on how good each term is at discriminating the relevant vs. irrelevant candidates. First, we demonstrate the efficacy of the distributed representations on the CLEF-IP 2010 dataset where we achieve significant improvement of 4.6% in recall over the keyword search baseline. Second, we simulate interactivity to demonstrate the efficacy of the proposed interactive term weighting mechanism. Simulation results show that we can achieve extra 2-12% improvement in mean av-

erage precision from one interaction iteration outperforming previous semantic and interactive patent retrieval methods.

9.1 Introduction

As mentioned in Chapter 8, patent retrieval is a challenging task. Patents are lengthy metadata rich documents. And patent queries, on the other hand, can be a complete multi-page patent application. These features make traditional information retrieval methods used for Web or ad hoc search inappropriate or at least of limited applicability to patent retrieval. We also highlighted in Chapter 8 that neither keyword-based or semantic-based methods has acceptable performance. Therefore, few interactive methods were proposed to better discriminate relevant vs. irrelevant query terms based on user feedback [62].

In this chapter, we present a novel interactive framework for patent retrieval based on distributed representations of concepts and entities identified in patents text. Offline, we jointly learn the embeddings of words, concepts, patent documents, and patent classes in the same semantic space. We then use the learned embeddings to generate multiple vector-based representations of the topic patent query and its prior art candidates. Given a topic patent, we find its prior art through two steps: 1) candidate generation through keyword search, favoring recall, and 2) candidate reranking through an ensemble of semantic similarities computed from the vector representations, favoring precision. Empirical evaluation of this automated retrieval scheme on the CLEF-IP 2010 dataset shows its efficacy over keyword search where we get 4.6% improvement in recall@100.

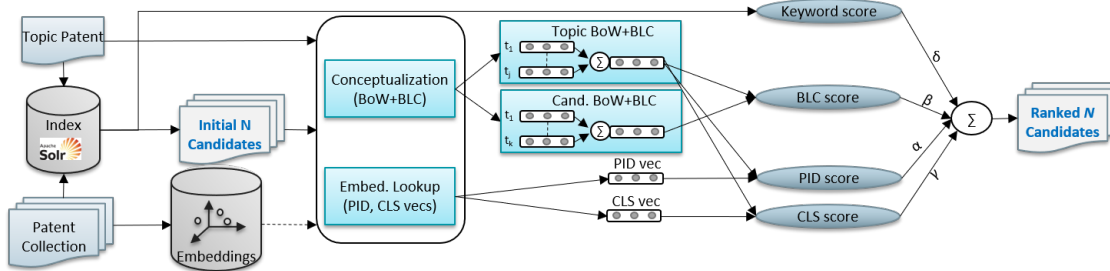


Figure 20: Automated reranking of prior art candidates using multiple scoring functions based on embeddings of words, concepts, patent documents, and patent classes.

In addition, we propose an effective query reformulation and term weighting mechanism based on interactive relevance feedback. We model term weighting as a supervised feature selection problem where term weights are assigned based on how good each term is at discriminating the relevant vs. irrelevant candidates obtained from user feedback. Our interaction mechanism is more practical and realistic than the one proposed by Golestan Far et al. [62]. We ask the user to annotate hits in the top n results as relevant/irrelevant, while in Golestan Far et al. [62] the user is restricted to annotate only relevant candidates which might appear very deep in the candidates list.

We simulate this interactive term weighting mechanism to demonstrate its effectiveness over the best performer on the CLEF-IP 2010 competition; PATATRAS [114]. Our simulation results show that we can outperform PATATRAS performance with only 1 annotated candidate regardless of whether it is relevant or not. It is worth mentioning that similar results have been presented in Golestan Far et al. [62], but with restricting the user to annotate 1 relevant candidate which again might require the user to navigate through several candidates⁸³.

⁸³Golestan Far et al. [62] figure 4 shows that ~ 750 of 1281 test queries have the 1st relevant candidate among the top 10 ($\sim 59\%$ chance).

9.2 Preprocessing and Offline Operations

9.2.1 The Search Index

As shown in Figure 20, automated vector-based retrieval starts by searching for an initial set of N candidates. For this purpose, we build a search index of the target candidate patents collection using Apache Solr. For each candidate patent, we index its *Id*, *title*, *abstract*, *description*, *claim1*, *claims*, and *IPC classification codes*. During candidate set generation, we use *title*, *abstract*, *description*, and *claims* of the topic patent and search all candidate fields except the IPC codes. We give equal weight to all the fields during search.

9.2.2 Text Conceptualization

By concepts/entities we mean single or multiword expressions which denote an idea, object, or event along with its characteristics. In the context of text mining, One flavor of text conceptualization works by extracting basic level concepts (BLC) from the input text by identifying mentions of those concepts and mapping them to entries in target Knowledge Base (KB). In this work, our concept space is defined by all *Wikipedia* article titles. We perform conceptualization by moving sliding windows of different sizes on the input patent text. Each window of size n will produce n -gram tokens which are then matched to a *Wikipedia* concept (article title) and replaced by unique Id.

Conceptualization has two main advantages: 1) concepts with different surface forms would be mapped to a single unique canonical form (e.g., *Solar cell*, *Photovoltaic cell*, *PV cell*), and 2) concept mentions of arbitrary length would be mapped to

unique Ids and therefore a single vector would be learned for each concept rather than each word of the concept expression (as described in Section 9.2.3). This is important for concepts whose meaning is different from the compositional semantics of its individual words (e.g., *rare earth element*). As shown in Figure 20, the output of text conceptualization is the union of the Bag-of-Words (BoW) and identified concept mentions (BLC) in the input patent text.

9.2.3 Learning Distributed Representations

Our framework adapts skip-gram [139], the popular local context window method, to jointly learn vector representations (embeddings) of words, concepts, patent documents, and patent classes in the same semantic space. By embedding all these structures in one space, we could measure the similarity between pairs of words, concepts, documents, and classes and between combinations of them using a proper similarity measure (e.g., *cosine*).

9.2.3.1 Word & Concept Vectors

We utilize the candidate patents collection as the input corpus. After all concept mentions are identified using text conceptualization, We train the skip-gram model to jointly learn the embeddings of both words and concepts using concept mentions. We apply the exact learning approach proposed for the *Concept Raw Context* model (CRX) introduced in Section 6.3.2. As described earlier, given a patent corpus of V words w_1, w_2, \dots, w_V . We iterate over the corpus identifying words and concept mentions and thus generating a sequence of T tokens t_1, t_2, \dots, t_T where $T < V$ (as multiword concepts will be counted as one token). Afterwards we train the skip-gram

aiming to maximize:

$$\mathcal{L}_t = \frac{1}{T} \sum_{i=1}^T \sum_{-s \leq j \leq s, j \neq 0} \log p(t_{i+j}|t_i) \quad (21)$$

where s is the context window size. Here, t_i is the target token which would be either a word or a concept mention, and t_{i+j} is a surrounding context word or concept mention.

9.2.3.2 Patent Documents Vectors

We learn unique vectors for each patent document with the objective to maximize the ability of predicting words/concepts appearing in the document given the patent vector. Therefore, contexts are generated as pairs of (t, pid_i) where t is a term (word/concept) appearing in a target patent document p_i whose Id is pid_i in the candidates collection C . Under this representation, our training objective would be maximizing:

$$\mathcal{L}_p = \frac{1}{|C|} \sum_{i=1}^{|C|} \sum_{t \in p_i} \log p(t_j|pid_i) \quad (22)$$

9.2.3.3 Patent Class Vectors

We learn unique vectors for each patent class. Patent classes are important in patent retrieval as they are assigned according to the patent technical features. Therefore, they can be used for soft filtering; to limit the scope of search to few class codes rather than searching through irrelevant technological fields. Our objective is to maximize the ability of predicting terms appearing in all the patents that belong to a target class given the class vector. Therefore, contexts are generated as pairs of (t, c) where t is a term (word/concept) appearing in a given patent document p which c is

one of its class codes CLS_p . Under this representation, our training objective would be maximizing:

$$\mathcal{L}_c = \frac{1}{|C|} \sum_{i=1}^{|C|} \sum_{t \in p} \sum_{c \in CLS_p} \log p(t|c) \quad (23)$$

During training, we train the embedding model to jointly maximize $\mathcal{L} = \mathcal{L}_t + \mathcal{L}_p + \mathcal{L}_c$ which is estimated using the softmax function.

As the patents vocabulary is typically full of jargon and user defined concepts, we start with the CRX vectors (described in Section 6.3.2); our pretrained concept embeddings model which utilizes *Wikipedia*.

9.3 Automated Vector-based Retrieval

Figure 20 shows the process of automated retrieval our framework. At a high level, given a topic patent, we retrieve an initial set of N candidates using keyword search from the Solr index. Then we create a vector representation for the topic patent and each candidate from the words and concept mentions in their corresponding text through conceptualization. We also generate another two vectors for each candidate through embedding lookup; one for the candidate patent document and the other for its class. After generating all the vectors, we compute similarity scores between the topic vector and each of the three vectors of each candidate. This will generate three scores which are then combined with the keyword search score to obtain the overall candidate relevancy score which is used to rank the N candidates. Below, we describe these steps in detail.

9.3.1 Vector Generation

In this step we generate continuous vectors for the topic patent and each of its prior art candidates. The BLC vector (\mathbf{v}_{blc}) is created from the weighted sum of the embeddings of all words and concepts in the patent text. We use the normalized term frequency (tf) as the initial term weight. Formally, given a patent whose text contains set of terms T , then $\mathbf{v}_{blc} = \sum_{i=1}^T w_i * lookup(t_i)$ where t_i is a word or a concept whose normalized tf is w_i and $lookup(.)$ retrieves the vector of its input from the learned embedding space.

We generate two other vectors for each candidate patent. First, the PID vector which corresponds to the vector learned for the whole patent document. It is obtained by $\mathbf{v}_{pid} = lookup(pid)$. Second, the CLS vector which corresponds to the vector of that patent class, and is obtained by $\mathbf{v}_{cls} = lookup(cls)$.

9.3.2 Candidate Scoring and Reranking

As mentioned earlier, the initial prior art candidates are obtained by keyword search. After generating the vectors of the topic patent and its candidates, we compute multiple semantic similarity scores which are then combined to produce the final relevancy score of each candidate to the topic patent. All scores utilize the cosine measure between pairs of vectors (\mathbf{u}, \mathbf{v}) as $cos(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|}$. In all below scores, \mathbf{u}_{blc} is the BLC vector of the topic patent, and \mathbf{v} is one of the vectors of a prior art candidate.

1. **BLC Score:** It is computed as $s_{blc} = cos(\mathbf{u}_{blc}, \mathbf{v}_{blc})$. It captures the fine-grained similarities between the two BLC vectors.

2. **PID Score:** It is computed as $s_{pid} = \cos(\mathbf{u}_{blc}, \mathbf{v}_{pid})$. It captures the coarse-grained similarities between the topic BLC vector and the whole candidate document.
3. **CLS Score:** It is computed as $s_{cls} = \cos(\mathbf{u}_{blc}, \mathbf{v}_{cls})$. It captures the similarity between the topic BLC vector and the high-level technical features of the candidate patent embedded in its class vector.
4. **Ensemble Scoring:** Finally, we combine the three scores with the normalized keyword search score (s_{kw}) through weighted sum to produce the final relevancy score of each candidate as $s = \alpha * s_{blc} + \beta * s_{pid} + \gamma * s_{cls} + \delta * s_{kw}$, where $\alpha + \beta + \gamma + \delta = 1$ and are tuned empirically.

9.3.3 Why Concept-based Distributed Representations?

The motivation behind using concept-based distributed representations for patent texts stems from the nature of the patent queries which tend to be long (with several hundreds or thousands of search terms). Therefore, we would expect repetitions of more important concepts vs. less important ones in the query. Turning queries into vectors allows us to focus on the important (frequent) concepts as the contribution of their corresponding dimensions to the overall similarity would be high favoring precision. On the other hand, less important (frequent) concepts would still contribute to the overall similarity but with much less magnitude favoring recall. The vector representation would also allow us to overcome the vocabulary mismatch problem as we match dimensions rather than symbols during similarity scoring. Last but not least, the vectors are fixed-size and thus computations would be much faster in the

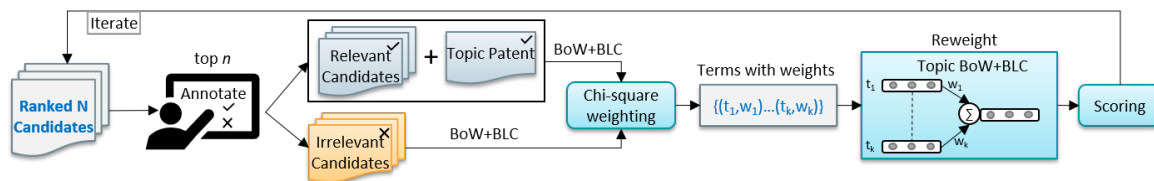


Figure 21: Interactive term weighting and scoring. Terms are weighted according to their relevancy in discriminating relevant vs. irrelevant candidates annotated by user.

vector space.

9.4 Interactive Relevance Feedback

As we will show in the evaluation section and indicated by previous studies [62], query reformulation (QRE) by means of expansion, removal, or reweighting of relevant/irrelevant terms could significantly boost the performance of patent retrieval. However, automated QRE fails to fully identify the significance of each term motivating the need for interactive QRE. Our framework embraces interactive relevance feedback for QRE. Inspired by Debole and Sebastiani [41], we model term weighting as a supervised feature selection problem where term weights are assigned based on how good each term is at discriminating the relevant vs. irrelevant candidates obtained from user feedback. Figure 21 shows the process of interactive QRE.

Our interaction mechanism is similar to the technology assisted review protocol [37]. After candidate reranking, the user is asked to annotate the top n candidates as either relevant or irrelevant to the topic patent. We then employ the chi-square (χ^2) statistic for term weighting considering the topic patent + the annotated relevant candidates as the +ve samples, while the annotated irrelevant ones as the -ve samples. After that, we create a modified \mathbf{v}_{blc} for the topic patent considering only those terms t in the topic patent and any of the annotated relevant candidates along with their

chi-square weights w_i such that $\mathbf{v}_{blc} = \sum_{i=1}^T w_i * lookup(t)$. The modified \mathbf{v}_{blc} is used to compute the ensemble scores and rerank the candidates. This process is iterated until the user is satisfied with the results.

We argue that our proposed user interaction mechanism is more practical than Golestan Far et al. [62]. In Golestan Far et al. [62] the user is required to annotate the relevant results only. However, this might be impractical as in patent retrieval it is usually expected that many relevant results appear late in the result set and therefore the user effort would be proportional to rankings of these relevant results. Our mechanism, alternatively, doesn't require the user to dig deep in the candidates list as we require the annotations of the top n candidates, therefore the user effort is proportional to n and independent from the relevant hits rankings. On the other hand, our proposed mechanism exploits both relevant and irrelevant hits as the user go through the candidates list. In case of no relevant candidates in the top n , we can still use the topic query as a relevant hit and apply chi-square weighting. In case of no irrelevant candidates in the top n , we can fall back to our normalized *tf* weighting expanding the topic patent terms with other terms from the annotated relevant ones.

9.5 Performance Evaluation

9.5.1 Experimental Setup

We evaluate our framework on the CLEF-IP 2010 benchmark dataset⁸⁴ which contains ~ 2.6 million patent documents. Similar to Golestan Far et al. [62], we considered only 1286 queries (topic patents) which has at least one relevant document whose *ti-*

⁸⁴<http://www.ifs.tuwien.ac.at/~clef-ip/>

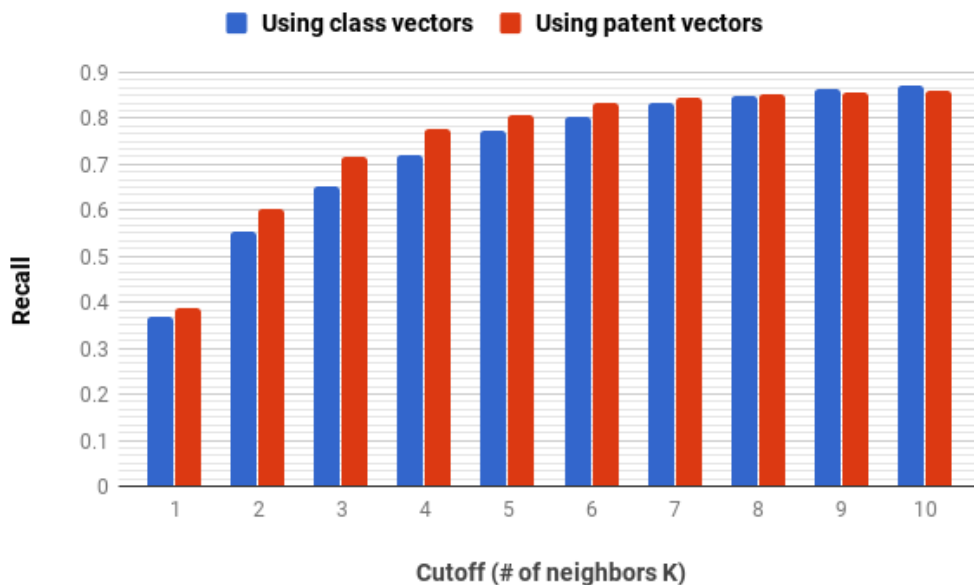


Figure 22: Classification recall with different number of neighbors (k) when classifying the CLEF-IP 2010 topic patents using the learned class codes vectors (blue) and patent document vectors (red).

tle, *abstract*, *description*, and *claims* in English. During keyword search, we set the number of initial candidates N to 1000. To make our results comparable to previous studies [62, 114], we perform IPC filtering during keyword search; keeping only candidates that share at least one IPC class with the topic query. We experimentally set $\alpha = 0.2, \beta = 0.4, \gamma = 0.125, \delta = 0.275$. We simulate user interactions by automatically annotating the top n candidates from the vector-based reranking using the true relevance judgments.

9.5.2 Are the Learned Vectors Meaningful?

In order to assess the quality of the learned vectors, we performed an intrinsic evaluation using these vectors to predict the IPC classification code(s) of each of the 1286 topic patents. We performed two experiments: 1) classification using the vectors

Table 38: Vector-based patent retrieval with interaction.

	Recall@100
Keyword baseline	41.9
PATATRAS [114]	46.7
Vector-based reranking	<u>46.7</u>
top 1 annotated	47.3
top 5 annotated	48.1
top 10 annotated	49.3

of the IPC class codes (\mathbf{v}_{cls}), and 2) classification using the PID vector of each of the topic patent candidates (\mathbf{v}_{pid}). In both cases, we first generate the BLC vector for each topic patent (\mathbf{u}_{blc}), and then use a k -nearest neighbors (k NN) classifier to predict the patent class(es). Each topic patent is assigned to k classes whose vectors (\mathbf{v}_{cls}) are most similar to \mathbf{u}_{blc} (when using class vectors), or the k classes of the candidates whose document vectors (\mathbf{v}_{pid}) are most similar to \mathbf{u}_{blc} (when using patent vectors). Figure 22 shows classification recall scores when varying the number of neighbors (k) between 1 and 10. As we can notice, the classification performance using the patent vectors is generally better compared to using the class vectors. Overall, the results are very promising, especially when $k > 1$ (cf. Table 5 recall scores), and show that the learned vectors encode meaningful representations of the patent documents as well as the classification codes.

9.5.3 Retrieval Results

Table 38 shows the performance of our system compared to PATATRAS [114], a patent retrieval system with significant preprocessing⁸⁵ and sophisticated use of patent

⁸⁵PATATRAS extracts some relevant patents from citations in the topic patent description using regex. This preprocessing step contributes up to 8% of their recall.

Table 39: MAP and PRES scores of vector-based patent retrieval compared to previous methods ordered by PRES score..

	MAP	PRES
Wang and Lin [206]	0.10	0.40
Mahdabi et al. [130]	0.12	0.49
Magdy and Jones [123]	0.14	0.49
Tannebaum et al. [196]	0.14	0.51
Bouadjenek et al. [18]	0.13	0.55
Vector-based reranking	0.14	0.63

Table 40: Percent improvements after the 1st interaction iteration over automatic vector-based reranking.

	Recall@5	Recall@10	Recall@50	Recall@100	MAP
Vector-based reranking	12.7	18.6	36.7	46.7	13.7
top 1 annotated	+1.4	+1.0	+1.2	+0.6	+1.9
top 5 annotated	+5.4	+4.2	+2.1	+1.5	+7.4
top 10 annotated	+9.4	+7.6	+3.2	+2.6	+11.6

metadata. As we can see, the automated vector-based reranking achieves equal performance to PATATRAS and improves recall by 4.6% compared to the keyword baseline demonstrating the usefulness of the learned distributed representations. Interactive QRE improves performance even more; we can outperform PATATRAS performance if the user annotates the first result from automated reranking as relevant or irrelevant. Table 39 shows the Mean Average Precision (MAP) and Patent REtrieval Score (PRES) of our method compared to previous keyword [130, 206], pseudo relevance feedback [18], and semantic-based methods [123, 196]. Our vector-based reranking gives very competitive score in terms of MAP and significantly outperforms all these methods in terms of PRES.

Annotating more results generally improves the performance. Table 40 shows per-

Table 41: Evaluation results of our vector-based reranking when coupled with the interactive relevance feedback mechanism of Golestan Far et al. [62]. (r is number of user annotated relevant candidates).

	MAP		Recall@100	
	$r=1$	$r=3$	$r=1$	$r=3$
Golestan Far et al. [62]*	28.8	36.9	47.9	54.7
Vector-based reranking + relevants	30.5	52.9	51.2	60.5

*Results from Golestan Far et al. [62] considering $\tau = 0$

cent improvements in recall at different ranks as well as MAP scores after *one* interaction iteration. As we can notice, with minimum user interaction effort (annotating 1 result), we can achieve extra 1.9% increase in MAP score. As the user annotates more results (5 and 10), the MAP score improves significantly (by 7.4% and 11.6% respectively). It is worth repeating that the user will be asked to annotate only the top n results regardless of their relevancy to the topic query.

Table 41 shows the results of our vector-based reranking compared to Golestan Far et al. [62]. For fair comparison, we report the results considering the user annotating relevant candidates only as in Golestan Far et al. [62]. Generally, we get more improvements as the user annotates more relevant hits. Our system gives much better results in terms of both Recall and MAP scores than Golestan Far et al. [62].

To better demonstrate the significance of our interaction and term weighting mechanism, we performed an experiment where we simulated the user annotating the top n candidates ($n = \{1,2,3,4,5,10\}$) from the vector-based reranking and then iterate over the new ranked list multiple iterations ($iter\# = \{1,2,3\}$). Figure 23 shows how Recall@K $\{K=5,10,50,100\}$ and MAP scores improve with increasing n and $iter\#$. As we can notice, significant improvement is achieved after $iter\#1$ with diminishing

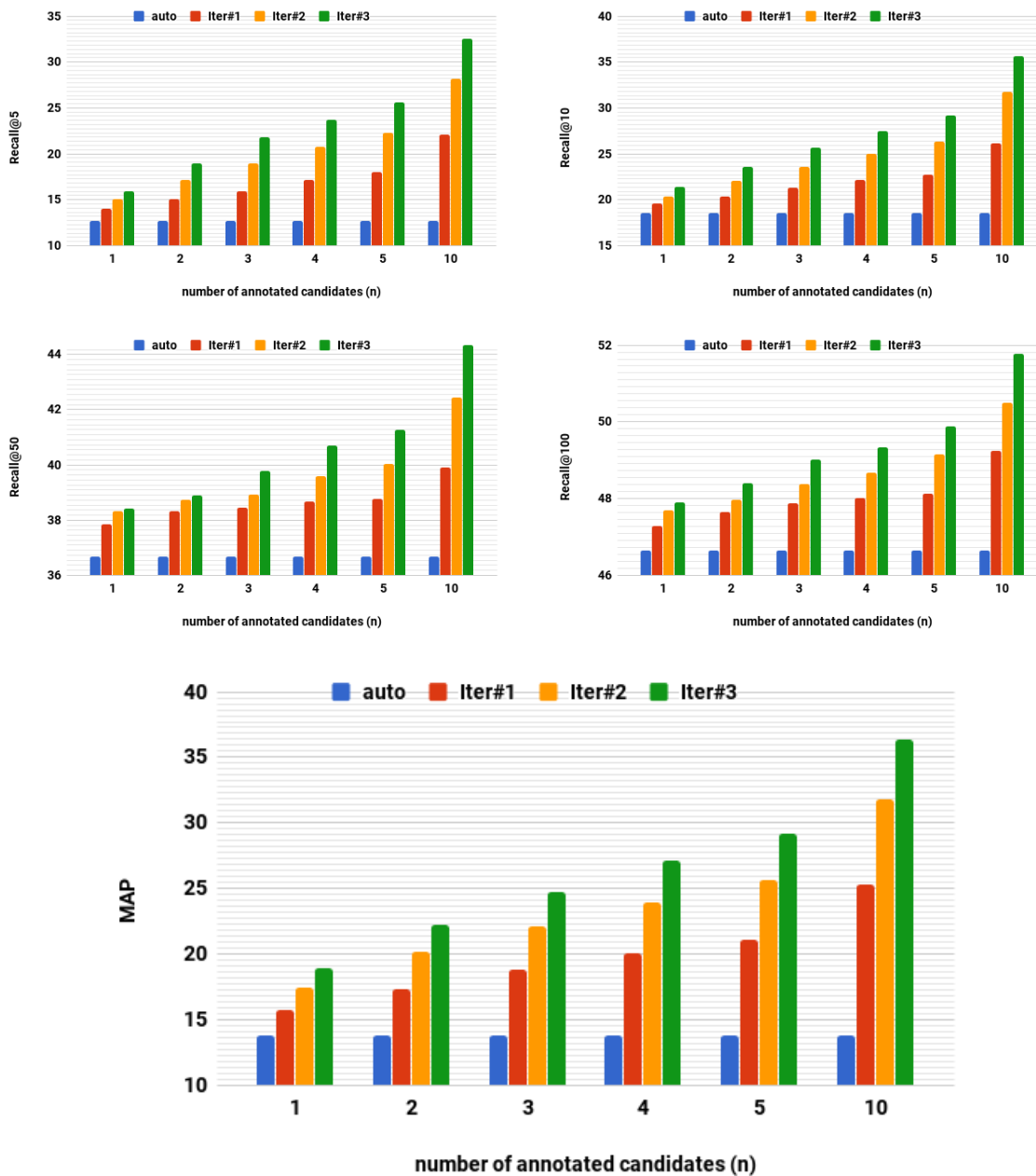


Figure 23: MAP & Recall @ different K of simulated interactive relevance feedback when varying the number of annotated hits n . Iter# is the number of simulated interaction iterations.

return as we iterate. We think this is because the diversity of vocabulary in the pool of candidate terms for chi-square weighting decreases as we iterate on the top hits causing weights to stabilize after 1 or 2 iterations. When n is relatively large (5 or 10),

the diversity keeps increasing and thus the magnitude of improvements is relatively higher with more iterations. We can also notice that user effort is proportional to n making our interaction mechanism more practical.

9.6 Conclusion

In this chapter, we presented a novel effective interactive framework for patent retrieval. Our framework is generic and can accept non-patent queries as well. We support human-in-the-loop through soliciting user feedback with reasonable effort. Under the hood, we utilize chi-square statistic to learn proper term weights and subsequently perform query reformulation to promote more relevant results and demote irrelevant ones. The proposed framework efficiently computes multiple similarity scores which capture semantic similarities at different levels (words, concepts, documents, and categories). Empirical results show superior performance of our system compared to previous fully automated keyword, semantic, and interactive methods.

CHAPTER 10: CONCLUSION AND FUTURE DIRECTIONS

The objective of this thesis was to improve the analysis and retrieval of textual data especially technical texts (e.g., patents, scientific literature...etc) using concept-based representations. We showed through empirical evaluation that: 1) significant performance improvements can be achieved using our novel concept-based representations with both long technical text (patents) and short text (search queries), 2) our concept-based representations greatly facilitate interactive and visual analysis of technical text, and 3) the proposed conceptual representations are generic and applicable to many academic benchmark datasets where we achieve superior state-of-the-art performance.

First, we presented a simple and efficient knowledge-based technique for reducing the dimensionality of the bag of n-grams model. Using our unsupervised technique, we achieved 13-fold reduction in the number of bigram features and 1.7% increase in classification accuracy over the bag-of-words baseline.

Second, we addressed the challenge of short text representation. We created an ensemble of contextual, knowledge-based, and lexical features for short texts. Evaluation results showed superior performance (97% micro-averaged F1 score).

Third, we presented Mined Semantic Analysis (MSA), utilizing unsupervised data mining techniques in order to discover concept-concept associations. These associations are used subsequently to enrich the bag-of-concepts representation of the given

text. Quantitative evaluation of MSA measuring text semantic similarity showed its superior performance. Additionally, we demonstrated the usability of MSA by implementing a Web-based semantic-driven visual and interactive framework for patent analytics.

Fourth, we proposed a neural-based model to learn distributed representations (embeddings) of concepts and entities from their mentions in encyclopedic knowledge bases. The model is space and computationally efficient; it overcomes the concept mismatch problem; it is expressive and interpretable.

Finally, we proposed a novel interactive framework for patent retrieval. Using the distributed representations on patents prior art search, we achieved significant improvement of 4.6% in recall. Simulation results of our proposed interaction mechanism showed that we can achieve extra 1.9% to 11.6% improvement in mean average precision from one interaction iteration, outperforming previous semantic and interactive patent retrieval methods.

The work done so far focused on improving the semantic representation of text structures as a prototype to enhanced text analysis and retrieval. The presented research in this thesis targeted three perspectives of the semantic representation: efficiency, effectiveness, and usability. The methods and ideas proposed for each perspective have the potential for different future work including extensions to those methods, and evaluation of them on other related tasks and datasets. For example:

- Our work on knowledge-based dimensionality reduction (Chapter 2) opens the doors for more ideas and hypotheses that need empirical assessment. For exam-

ple, it was reported in Benzineb and Guyot [7] that adding unigrams features from patent description text improves accuracy, though it introduces computational complexities due to the large number of generated features. The intuitive aspect of our approach comes into play when answering a question of which field or combination of fields of patent documents are more pertinent as source of classification features (e.g., *abstract*, *description*, *claims*). Our approach will be of great relevance, especially when incorporating larger lexical features (trigrams). We can also utilize the rich knowledge sources differently by including all concepts with links from Wikipedia articles as candidate features which might reduce the semantic ambiguity and the size of unigrams. Another possible extension of our work is probing the impact of leveraging other structured knowledge bases like DBPedia⁸⁶ and Freebase⁸⁷ for feature selection. These sources might allow more sophisticated linguistic and meta-features to be extracted at low cost.

- Patent classification is a challenging problem. On the one hand, the class distribution is very skewed (see Figure 4). On the other hand, the patent taxonomy changes over time where new categories and subcategories are continuously created. These two facts motivate the need for robust classification models that can work with no or small number of labeled examples. And this is where the dataless classification protocol might be useful. A possible future direction is to utilize and evaluate the proposed dense Bag-of-Concepts (BoC) representations

⁸⁶<http://dbpedia.org/>

⁸⁷<https://www.freebase.com/>

for the task of automated patent classification.

- We demonstrated the effectiveness of ensemble representations for entity type recognition of search queries (Chapter 3). Initially, we targeted the four most important categories (*Company*, *School*, *Job title*, and *Skill*) in the recruitment domain. One direct extension is to expand these categories to include other important types such as *location*, or capture fine-grained aspects such as *job level* (e.g., entry, junior, senior...etc), and *skill type* (e.g., social, soft, professional, language...etc). Concept-based representation of search queries using MSA could be another feature added to the ensemble representation.
- As obtaining labeled data is typically costly and labor intensive. One possible application of our concept embedding model is through modeling the entity type recognition problem as a concept categorization or dataless classification problem, where entity types would represent categories and target entities would represent unlabeled instances. Under both cases we could leverage our concept embeddings model to infer entity types. For example, under the concept categorization assumption, we would obtain entity and category embeddings directly from the model. While, under the dataless classification assumption, we would first generate BoC representations for entities and categories and then use the concept embeddings model to obtain dense BoC representations. The dataless classification assumption would be more adequate for those entities and categories that do not have representations in our model. The concept categorization assumption would more efficient as it does not require generating BoC

vectors in the first place.

- In Chapter 9 we presented an effective concept-based distributed representation model for patent retrieval. We showed the value of coupling this model with simple yet effective interactive relevance feedback method for term weighting. One possible venue for future work is exploring and validating these models on other text retrieval tasks and/or other patent benchmark datasets. Another extension is integrating the relevance feedback mechanism with the visual framework presented in Chapter 5 and exploring the usefulness of that mechanism on other patent analysis use cases.
- The proposed concept embedding model learns concept vectors from their mentions in Wikipedia (Chapter 6). A straightforward extension would be to learn multilingual concept embeddings leveraging other Wikipedias⁸⁸. The learned embeddings could help creating multilingual concept-based KBs or curating existing ones by applying the concept categorization protocol as described in Section 6.4.2.
- Learning from raw concept mentions makes our approach applicable to other open domain and domain specific free-text corpora as well. This can be accomplished by firstly wikifying the text and then learning from concept mentions. For example, we can learn concept embeddings of one of the main patent classes (e.g., *Telecommunication*). And utilize the learned embeddings for better analysis of patent documents (e.g., concept-based exploration and search).

⁸⁸https://en.wikipedia.org/wiki/List_of_Wikipedias#List

- Co-training is another application area where concept embeddings might be useful. The idea is to use the pretrained concept and word vectors to initialize the embedding layer of a Deep Neural Network (DNN) architecture, rather than learning the embeddings from scratch. This idea has proven effectiveness when applied to sentiment analysis [179] and political ideology detection [91]. We performed an experiment to investigate the value of using the CRX model's concept and word vectors (described in Section 6.3.2) to initialize the embedding layer of a Recurrent Neural Network (RNN) for building a language model [217] on the CoNLL-2003 dataset [197]. The results show that, we can achieve 3.8% improvement in perplexity compared to initializing the embedding layer with word vectors only. This result along with the results on entity type identification with semantic parsing (described in Section 7.3.3), demonstrate the need for adopting entity-aware and concept-aware training of DNNs for enhanced representations and better performance.
- Evaluating the concept embedding models introduced in Chapter 6 and Chapter 7 on other tasks such as measuring lexical and document semantic similarity.
- Incorporating semantic-role-labeling based representation with the BoC has proven effectiveness for event detection [153]. However, generating a BoC vector of a few hundred dimensions is costly, and becomes even more costly when generated for each unit of the semantic parse. One possible future direction it to explore the viability of semantic and syntactic parsing with the dense BoC representations using concept embeddings. As we demonstrated in Chapter 6,

the cost of creating a concept vector of few dimensions would be much less with dense BoC compared to the sparse BoC. Thereafter, similarity calculations could be through comparing the dense BoC of syntactically/semantically identical constituents.

- Exploring DNN architectures for patent retrieval and classification is another future direction. There has been so much work applying Convolutional Neural Networks (CNN) to classification and categorization of short and medium length text (e.g., tweets, movie reviews). For information retrieval, there have been so much work on sentence modeling for article recommendation. None of these approaches explore the applicability of such architectures with long documents such as patents. Interestingly, our concept embeddings could act as an extra channel in the CNN input layer besides other word-based embeddings such as Word2Vec and Glove.

REFERENCES

- [1] Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. A study on similarity and relatedness using distributional and wordnet-based approaches. In *Proceedings of Human Language Technologies: NAACL 2009*, pages 19–27. ACL, 2009.
- [2] Rakesh Agrawal, Ramakrishnan Srikant, et al. Fast algorithms for mining association rules.
- [3] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. In *ACM SIGMOD Record*, volume 22, pages 207–216. ACM, 1993.
- [4] Bashar Al-Shboul and Sung-Hyon Myaeng. Wikipedia-based query phrase expansion in patent class search. *Information Retrieval*, 17(5-6):430–451, 2014.
- [5] Khalifeh AlJadda, Mohammed Korayem, Trey Grainger, and Craig Russell. Crowdsourced query augmentation through semantic discovery of domain-specific jargon. In *Big Data (Big Data), 2014 IEEE International Conference on*, pages 808–815. IEEE, 2014.
- [6] Khalifeh AlJadda, Mohammed Korayem, Camilo Ortiz, Trey Grainger, John A Miller, and William S York. Pgmhd: A scalable probabilistic graphical model for massive hierarchical data problems. In *Big Data (Big Data), 2014 IEEE International Conference on*, pages 55–60. IEEE, 2014.
- [7] Khalifeh AlJadda, Mohammed Korayem, and Trey Grainger. Improving the quality of semantic relationships extracted from massive user behavioral data. In *Big Data (Big Data), 2015 IEEE International Conference on*, pages 2951–2953. IEEE, 2015.
- [8] John R Allison, Mark A Lemley, and David L Schwartz. Understanding the realities of modern patent litigation. 2014.
- [9] John R Allison, Mark A Lemley, and David L Schwartz. Our divided patent system. *The University of Chicago Law Review*, pages 1073–1154, 2015.
- [10] Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al. *Modern information retrieval*, volume 463. ACM press New York, 1999.
- [11] Marco Baroni and Alessandro Lenci. Distributional memory: A general framework for corpus-based semantics. *Computational Linguistics*, 36(4):673–721, 2010.
- [12] Marco Baroni, Georgiana Dinu, and Germán Kruszewski. Don’t count, predict! a systematic comparison of context-counting vs. context-predicting semantic

- vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 238–247, 2014.
- [13] Shariq Bashir and Andreas Rauber. Improving retrievability of patents in prior-art search. In *Advances in Information Retrieval*, pages 457–470. Springer, 2010.
- [14] Ron Bekkerman and James Allan. Using bigrams in text categorization. Technical report, Technical Report IR-408, Center of Intelligent Information Retrieval, UMass Amherst, 2004.
- [15] Karim Benzineb and Jacques Guyot. Automated patent classification. In *Current challenges in patent information retrieval*, pages 239–261. Springer, 2011.
- [16] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.
- [17] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pages 2787–2795, 2013.
- [18] Mohamed Reda Bouadjenek, Scott Sanner, and Gabriela Ferraro. A study of query reformulation for patent prior art search with partial patent applications. In *Proceedings of the 15th International Conference on Artificial Intelligence and Law*, pages 23–32. ACM, 2015.
- [19] Pierre Boullier and Benoît Sagot. Efficient and robust lfg parsing: Sxlf. In *Proceedings of the Ninth International Workshop on Parsing Technology*, pages 1–10. Association for Computational Linguistics, 2005.
- [20] Sergey Brin and Lawrence Page. Reprint of: The anatomy of a large-scale hypertextual web search engine. *Computer networks*, 56(18):3825–3833, 2012.
- [21] Elia Bruni, Nam-Khanh Tran, and Marco Baroni. Multimodal distributional semantics. *J. Artif. Intell. Res.(JAIR)*, 49:1–47, 2014.
- [22] Alexander Budanitsky and Graeme Hirst. Evaluating wordnet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32(1):13–47, 2006.
- [23] José Camacho-Collados, Mohammad Taher Pilehvar, and Roberto Navigli. Nasari: a novel approach to a semantically-aware representation of items. In *Proceedings of NAACL*, pages 567–577, 2015.
- [24] José Camacho-Collados, Mohammad Taher Pilehvar, and Roberto Navigli. Nasari: Integrating explicit knowledge and corpus statistics for a multilingual representation of concepts and entities. *Artificial Intelligence*, 240:36–64, 2016.
- [25] Guihong Cao, Jian-Yun Nie, Jianfeng Gao, and Stephen Robertson. Selecting good expansion terms for pseudo-relevance feedback. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 243–250. ACM, 2008.

- [26] Yixin Cao, Lifu Huang, Heng Ji, Xu Chen, and Juanzi Li. Bridge text and knowledge by learning multi-prototype entity mention embedding. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1623–1633, 2017.
- [27] Jaime Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336. ACM, 1998.
- [28] David Carmel, Ming-Wei Chang, Evgeniy Gabilovich, Bo-June Paul Hsu, and Kuansan Wang. Erd’14: entity recognition and disambiguation challenge. In *ACM SIGIR Forum*, volume 48, pages 63–77. ACM, 2014.
- [29] Maria F Caropreso, Stan Matwin, and Fabrizio Sebastiani. Statistical phrases in automated text categorization. *Centre National de la Recherche Scientifique, Paris, France*, 47, 2000.
- [30] Diego Ceccarelli, Claudio Lucchese, Salvatore Orlando, Raffaele Perego, and Salvatore Trani. Learning relatedness measures for entity linking. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 139–148. ACM, 2013.
- [31] Ming-Wei Chang, Lev-Arie Ratinov, Dan Roth, and Vivek Srikumar. Importance of semantic representation: Dataless classification. In *AAAI*, volume 2, pages 830–835, 2008.
- [32] Yen-Liang Chen and Yu-Ting Chiu. An ipc-based vector space model for patent retrieval. *Information Processing & Management*, 47(3):309–322, 2011.
- [33] Ying Chen, Scott Spangler, Jeffrey Kreulen, Stephen Boyer, Thomas D Griffin, Alfredo Alba, Amit Behal, Bin He, Linda Kato, Ana Lelescu, et al. Simple: a strategic information mining platform for licensing and execution. In *Data Mining Workshops, 2009. ICDMW’09. IEEE International Conference on*, pages 270–275. IEEE, 2009.
- [34] Jennifer Chu-Carroll, James Fan, Nico Schlaefer, and Wlodek Zadrozny. Textual resource acquisition and engineering. *IBM Journal of Research and Development*, 56(3.4):4–1, 2012.
- [35] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM, 2008.
- [36] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537, 2011.

- [37] Gordon V Cormack and Maura R Grossman. Evaluation of machine-learning protocols for technology-assisted review in electronic discovery. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 153–162. ACM, 2014.
- [38] Steve Cronen-Townsend, Yun Zhou, and W Bruce Croft. Predicting query performance. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 299–306. ACM, 2002.
- [39] Silviu Cucerzan. Large-scale named entity disambiguation based on wikipedia data. In *EMNLP-CoNLL*, volume 7, pages 708–716, 2007.
- [40] Dirk Czarnitzki, Katrin Hussinger, and Bart Leten. The market value of blocking patents, 2010.
- [41] Franca Debole and Fabrizio Sebastiani. Supervised term weighting for automated text categorization. In *Text mining and its applications*, pages 81–97. Springer, 2004.
- [42] Eva D’hondt and Suzan Verberne. Clef-ip 2010: Prior art retrieval using the different sections in patent documents. In *CLEF (Notebook Papers/LABs/Workshops)*, 2010.
- [43] Eva D’hondt, Suzan Verberne, Cornelis Koster, and Lou Boves. Text representations for patent classification. *Computational Linguistics*, 39(3):755–775, 2013.
- [44] Li Dong and Mirella Lapata. Language to logical form with neural attention. *arXiv preprint arXiv:1601.01280*, 2016.
- [45] Ofer Egozi, Shaul Markovitch, and Evgeniy Gabrilovich. Concept-based information retrieval using explicit semantic analysis. *ACM Transactions on Information Systems (TOIS)*, 29(2):8, 2011.
- [46] Daniel Eisinger, George Tsatsaronis, Markus Bundschuh, Ulrich Wieneke, and Michael Schroeder. Automated patent categorization and guided patent search using ipc as inspired by mesh and pubmed. *Journal of biomedical semantics*, 4(1):1, 2013.
- [47] Pavlos Fafalios and Yannis Tzitzikas. Exploratory professional search through semantic post-analysis of search results. In *Professional Search in the Modern World*, pages 166–192. Springer, 2014.
- [48] Mona Golestan Far, Scott Sanner, Mohamed Reda Bouadjeneq, Gabriela Ferraro, and David Hawking. On term selection techniques for patent prior art search. In *SIGIR’15: 38th International SIGIR Conference on Research and Development in Information Retrieval*, 2015.

- [49] Christiane Fellbaum. *WordNet: An Electronic Lexical Database*. Bradford Books, 1998.
- [50] Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppın. Placing search in context: The concept revisited. In *Proceedings of the 10th international conference on World Wide Web*, pages 406–414. ACM, 2001.
- [51] Atsushi Fujii. Enhancing patent retrieval by citation analysis. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 793–794. ACM, 2007.
- [52] Atsushi Fujii, Makoto Iwayama, and Noriko Kando. Overview of patent retrieval task at ntcir-4. In *NTCIR*, 2004.
- [53] Atsushi Fujii, Makoto Iwayama, and Noriko K. Overview of patent retrieval task at ntcir-5. In *In Proceedings of the Fifth NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering and Cross-lingual Information Access*, pages 269–277, 2005.
- [54] Atsushi Fujii, Makoto Iwayama, and Noriko Kando. Overview of the patent retrieval task at the ntcir-6 workshop. In *NTCIR*, 2007.
- [55] Evgeniy Gabrilovich and Shaul Markovitch. Overcoming the brittleness bottleneck using wikipedia: Enhancing text categorization with encyclopedic knowledge. In *AAAI*, volume 6, pages 1301–1306, 2006.
- [56] Evgeniy Gabrilovich and Shaul Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *IJCAI*, volume 7, pages 1606–1611, 2007.
- [57] Debasis Ganguly, Johannes Leveling, Walid Magdy, and Gareth JF Jones. Patent query reduction using pseudo relevance feedback. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 1953–1956. ACM, 2011.
- [58] Tong Gao, Jessica R Hullman, Eytan Adar, Brent Hecht, and Nicholas Diakopoulos. Newsviews: an automated pipeline for creating custom geovisualizations for news. In *Proceedings of the 32nd annual ACM conference on Human Factors in Computing Systems*, pages 3005–3014. ACM, 2014.
- [59] Abhishek Gattani, Digvijay S Lamba, Nikesh Garera, Mitul Tiwari, Xiaoyong Chai, Sanjib Das, Sri Subramaniam, Anand Rajaraman, Venky Harinarayan, and AnHai Doan. Entity extraction, linking, classification, and tagging for social media: a wikipedia-based approach. *Proceedings of the VLDB Endowment*, 6(11):1126–1137, 2013.

- [60] Anastasia Giachanou, Michail Salampanis, and Georgios Paltoglou. Multilayer source selection as a tool for supporting patent search and classification. *Information Retrieval Journal*, 18(6):559–585, 2015.
- [61] Julien Gobeill, Emilie Pasche, Douglas Teodoro, and Patrick Ruch. Simple pre and post processing strategies for patent searching in clef intellectual property track 2009. In *Multilingual Information Access Evaluation I. Text Retrieval Experiments*, pages 444–451. Springer, 2009.
- [62] Mona Golestan Far, Scott Sanne, Mohamed Reda Bouadjenek, Gabriela Ferraro, and David Hawking. On term selection techniques for patent prior art search. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 803–806. ACM, 2015.
- [63] Erik Graf and Leif Azzopardi. A methodology for building a patent test collection for prior art search. In *In Proceedings of the 2nd International Workshop on Evaluating Information Access, EVIA*, 2008.
- [64] Maura R Grossman and Gordon V Cormack. Technology-assisted review in e-discovery can be more effective and more efficient than exhaustive manual review. *Rich. JL & Tech.*, 17:11–16, 2011.
- [65] Jiafeng Guo, Gu Xu, Xueqi Cheng, and Hang Li. Named entity recognition in query. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 267–274. ACM, 2009.
- [66] Jacques Guyot, Karim Benzineb, Gilles Falquet, and Simple Shift. my-class: A mature tool for patent classification. In *CLEF (Notebook Papers/LABs/Workshops)*, 2010.
- [67] Guy Halawi, Gideon Dror, Evgeniy Gabrilovich, and Yehuda Koren. Large-scale learning of word relatedness with constraints. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1406–1414. ACM, 2012.
- [68] Bronwyn H Hall, Adam Jaffe, and Manuel Trajtenberg. Market value and patent citations. *RAND Journal of economics*, pages 16–38, 2005.
- [69] Xianpei Han and Le Sun. An entity-topic model for entity linking. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 105–115. Association for Computational Linguistics, 2012.
- [70] Xianpei Han, Le Sun, and Jun Zhao. Collective entity linking in web text: a graph-based method. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 765–774. ACM, 2011.

- [71] Tam Harbert. The law machine. *Spectrum, IEEE*, 50(11):31–54, 2013.
- [72] Dietmar Harhoff, Francis Narin, Frederic M Scherer, and Katrin Vopel. Citation frequency and the value of patented inventions. *Review of Economics and statistics*, 81(3):511–515, 1999.
- [73] Christopher G Harris, Steven Foster, Robert Arens, and Padmini Srinivasan. On the role of classification in patent invalidity searches. In *Proceedings of the 2nd international workshop on Patent information retrieval*, pages 29–32. ACM, 2009.
- [74] Christopher G Harris, Robert Arens, and Padmini Srinivasan. Comparison of ipc and uspc classification systems in patent prior art searches. In *Proceedings of the 3rd international workshop on Patent Information Retrieval*, pages 27–32. ACM, 2010.
- [75] Zellig S Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.
- [76] Mohammad Al Hasan, W Scott Spangler, Thomas Griffin, and Alfredo Alba. Coa: Finding novel patents through text analysis. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1175–1184. ACM, 2009.
- [77] Samer Hassan and Rada Mihalcea. Semantic relatedness using salient semantic analysis. In *AAAI*, 2011.
- [78] Shohei Hido, Shoko Suzuki, Risa Nishiyama, Takashi Imamichi, Rikiya Takahashi, Tetsuya Nasukawa, Tsuyoshi Idé, Yusuke Kanehira, Rinju Yohda, Takeshi Ueno, et al. Modeling patent quality: A system for large-scale patentability analysis using text mining. *Information and Media Technologies*, 7(3):1180–1191, 2012.
- [79] Djoerd Hiemstra, Stephen Robertson, and Hugo Zaragoza. Parsimonious language models for information retrieval. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 178–185. ACM, 2004.
- [80] Felix Hill, KyungHyun Cho, Sebastien Jean, Coline Devin, and Yoshua Bengio. Not all neural embeddings are born equal. *arXiv preprint arXiv:1410.0718*, 2014.
- [81] Thomas Hill and Paul Lewicki. *Statistics: Methods and Applications*. StatSoft, Inc, 2007.
- [82] Johannes Hoffart, Stephan Seufert, Dat Ba Nguyen, Martin Theobald, and Gerhard Weikum. Kore: keyphrase overlap relatedness for entity disambiguation. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 545–554. ACM, 2012.

- [83] Jian Hu, Gang Wang, Fred Lochovsky, Jian-tao Sun, and Zheng Chen. Understanding user's query intent with wikipedia. In *Proceedings of the 18th international conference on World wide web*, pages 471–480. ACM, 2009.
- [84] Po Hu, Minlie Huang, Peng Xu, Weichang Li, Adam K Usadi, and Xiaoyan Zhu. Finding nuggets in ip portfolios: core patent mining through textual temporal analysis. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 1819–1823. ACM, 2012.
- [85] Zhiting Hu, Poyao Huang, Yuntian Deng, Yingkai Gao, and Eric P Xing. Entity hierarchy embedding. In *Proceedings of The 53rd Annual Meeting of the Association for Computational Linguistics*, 2015.
- [86] Wen Hua, Zhongyuan Wang, Haixun Wang, Kai Zheng, and Xiaofang Zhou. Short text understanding through lexical-semantic analysis. In *Data Engineering (ICDE), 2015 IEEE 31st International Conference on*, pages 495–506. IEEE, 2015.
- [87] Hongzhao Huang, Larry Heck, and Heng Ji. Leveraging deep neural networks and knowledge graphs for entity disambiguation. *arXiv preprint arXiv:1504.07678*, 2015.
- [88] Ioana Hulpus, Narumol Prangnawarat, and Conor Hayes. Path-based semantic relatedness on linked data and its use to word and entity disambiguation. In *International Semantic Web Conference*, pages 442–457. Springer, 2015.
- [89] Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. Sensembed: Learning sense embeddings for word and relational similarity. In *ACL (1)*, pages 95–105, 2015.
- [90] Makoto Iwayama, Atsushi Fujii, Noriko Kando, and Akihiko Takano. Overview of patent retrieval task at ntcir-3. In *Proceedings of the ACL-2003 workshop on Patent corpus processing-Volume 20*, pages 24–32. Association for Computational Linguistics, 2003.
- [91] Mohit Iyyer, Peter Enns, Jordan Boyd-Graber, and Philip Resnik. Political ideology detection using recursive neural networks. In *Proceedings of the Association for Computational Linguistics*, pages 1113–1122, 2014.
- [92] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002.
- [93] Xin Jin, Scott Spangler, Ying Chen, Keke Cai, Rui Ma, Li Zhang, Xian Wu, and Jiawei Han. Patent maintenance recommendation with patent information network model. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, pages 280–289. IEEE, 2011.

- [94] Julia J Jürgens, Preben Hansen, and Christa Womser-Hacker. Going beyond clef-ip: The reality for patent searchers? In *Information Access Evaluation. Multilinguality, Multimodality, and Visual Analytics*, pages 30–35. Springer, 2012.
- [95] Junichi Kazama and Kentaro Torisawa. Exploiting wikipedia as external knowledge for named entity recognition. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 698–707, 2007.
- [96] Dongwoo Kim, Haixun Wang, and Alice H Oh. Context-dependent conceptualization. In *IJCAI*, pages 2330–2336, 2013.
- [97] Jungi Kim, In-Su Kang, and Jong-Hyeok Lee. Cluster-based patent retrieval using international patent classification system. In *Computer Processing of Oriental Languages. Beyond the Orient: The Research Challenges Ahead*, pages 205–212. Springer, 2006.
- [98] Steffen Koch, Harald Bosch, Mark Giereth, and Thomas Ertl. Iterative integration of visual insights during scalable patent search and analysis. *Visualization and Computer Graphics, IEEE Transactions on*, 17(5):557–569, 2011.
- [99] Kazuya Konishi. Query terms extraction from patent document for invalidity search. In *NTCIR*, 2005.
- [100] Mohammed Korayem, Camilo Ortiz, Khalifeh AlJadda, and Trey Grainger. Query sense disambiguation leveraging large scale user behavioral data. In *Big Data (Big Data), 2015 IEEE International Conference on*, pages 1230–1237. IEEE, 2015.
- [101] Ralf Krestel and Padhraic Smyth. Recommending patents based on latent topics. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 395–398. ACM, 2013.
- [102] Sayali Kulkarni, Amit Singh, Ganesh Ramakrishnan, and Soumen Chakrabarti. Collective annotation of wikipedia entities in web text. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 457–466. ACM, 2009.
- [103] Michal Laclavík, Marek Ciglan, Sam Steingold, Martin Seleng, Alex Dorman, and Stefan Dlugolinsky. Search query categorization at scale. In *Proceedings of the 24th International Conference on World Wide Web Companion*, pages 1281–1286. International World Wide Web Conferences Steering Committee, 2015.
- [104] Savio LY Lam and Dik Lun Lee. Feature reduction for neural network based text categorization. In *Database Systems for Advanced Applications, 1999. Proceedings., 6th International Conference on*, pages 195–202. IEEE, 1999.

- [105] Thomas K Landauer, Darrell Laham, Bob Rehder, and Missy E Schreiner. How well can passage meaning be derived without using word order? a comparison of latent semantic analysis and humans. In *Proceedings of the 19th annual meeting of the Cognitive Science Society*, pages 412–417. Citeseer, 1997.
- [106] Ken Lang. Newsweeder: Learning to filter netnews. In *Proceedings of the 12th international conference on machine learning*, pages 331–339, 1995.
- [107] Jean O Lanjouw, Ariel Pakes, and Jonathan Putnam. How to count patents and value intellectual property: The uses of patent renewal and application data. *The Journal of Industrial Economics*, 46(4):405–432, 1998.
- [108] M Lee, Brandon Pincombe, and Matthew Welsh. An empirical evaluation of models of text document similarity. Cognitive Science Society, 2005.
- [109] Peipei Li, Haixun Wang, Kenny Q Zhu, Zhongyuan Wang, and Xindong Wu. Computing term similarity by large probabilistic isa knowledge. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 1401–1410. ACM, 2013.
- [110] Yuezhong Li, Ronghuo Zheng, Tian Tian, Zhiting Hu, Rahul Iyer, and Katia Sycara. Joint embedding of hierarchical categories and entities for concept categorization and dataless classification. *arXiv preprint arXiv:1607.07956*, 2016.
- [111] Thomas Lin and Oren Etzioni. Entity linking at web scale. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, pages 84–88. Association for Computational Linguistics, 2012.
- [112] Shuang Liu, Fang Liu, Clement Yu, and Weiyi Meng. An effective approach to document retrieval via utilizing wordnet and recognizing phrases. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 266–272. ACM, 2004.
- [113] Yan Liu, Pei-yun Hseuh, Rick Lawrence, Steve Meliksetian, Claudia Perlich, and Alejandro Veen. Latent graphical models for quantifying and predicting patent quality. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1145–1153. ACM, 2011.
- [114] Patrice Lopez and Laurent Romary. Multiple retrieval models and regression models for prior art search. In *CLEF 2009 Workshop*, page 18p, 2009.
- [115] Patrice Lopez and Laurent Romary. Experiments with citation mining and key-term extraction for prior art search. In *CLEF 2010-Conference on Multilingual and Multimodal Information Access Evaluation*, 2010.
- [116] Mihai Lupu, Jimmy Huang, Jianhan Zhu, and John Tait. Trec-chem: large scale chemical information retrieval evaluation at trec. In *ACM SIGIR Forum*, volume 43, pages 63–70. ACM, 2009.

- [117] Mihai Lupu, John Tait, Jimmy Huang, and Jianhan Zhu. Trec-chem 2010: Notebook report. *Proceedings of TREC 2010*, 2, 2010.
- [118] Mihai Lupu, Katja Mayer, John Tait, and Anthony J Trippe. *Current challenges in patent information retrieval*, volume 29. Springer Science & Business Media, 2011.
- [119] Mihai Lupu, Jiashu Zhao, Jimmy Huang, Harsha Gurulingappa, Juliane Fluck, Marc Zimmermann, Igor V Filippov, and John Tait. Overview of the trec 2011 chemical ir track. In *TREC*, 2011.
- [120] Yuanhua Lv and ChengXiang Zhai. Adaptive relevance feedback in information retrieval. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 255–264. ACM, 2009.
- [121] Walid Magdy and Gareth JF Jones. Applying the kiss principle for the clef-ip 2010 prior art candidate patent search task. 2010.
- [122] Walid Magdy and Gareth JF Jones. Pres: a score metric for evaluating recall-oriented information retrieval applications. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 611–618. ACM, 2010.
- [123] Walid Magdy and Gareth JF Jones. A study on query expansion methods for patent retrieval. In *Proceedings of the 4th workshop on Patent information retrieval*, pages 19–24. ACM, 2011.
- [124] Walid Magdy, Johannes Leveling, and Gareth JF Jones. Exploring structured documents and query formulation techniques for patent retrieval. In *Multilingual Information Access Evaluation I. Text Retrieval Experiments*, pages 410–417. Springer, 2009.
- [125] Walid Magdy, Patrice Lopez, and Gareth JF Jones. Simple vs. sophisticated approaches for patent prior-art search. In *Advances in Information Retrieval*, pages 725–728. Springer, 2011.
- [126] Parvaz Mahdabi and Fabio Crestani. Learning-based pseudo-relevance feedback for patent retrieval. In *Multidisciplinary Information Retrieval*, pages 1–11. Springer, 2012.
- [127] Parvaz Mahdabi and Fabio Crestani. The effect of citation analysis on query expansion for patent retrieval. *Information Retrieval*, 17(5-6):412–429, 2014.
- [128] Parvaz Mahdabi and Fabio Crestani. Patent query formulation by synthesizing multiple sources of relevance evidence. *ACM Transactions on Information Systems (TOIS)*, 32(4):16, 2014.

- [129] Parvaz Mahdabi and Fabio Crestani. Query-driven mining of citation networks for patent citation retrieval and recommendation. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 1659–1668. ACM, 2014.
- [130] Parvaz Mahdabi, Mostafa Keikha, Shima Gerani, Monica Landoni, and Fabio Crestani. *Building queries for prior-art search*. Springer, 2011.
- [131] Parvaz Mahdabi, Shima Gerani, Jimmy Xiangji Huang, and Fabio Crestani. Leveraging conceptual lexicon: query disambiguation using proximity information for patent retrieval. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 113–122. ACM, 2013.
- [132] Massimiliano Mancini, Jose Camacho-Collados, Ignacio Iacobacci, and Roberto Navigli. Embedding words and senses together via joint knowledge-enhanced training. *arXiv preprint arXiv:1612.02703*, 2016.
- [133] Ronald J Mann and Marian Underweiser. A new look at patent quality: Relating patent prosecution to validity. *Journal of Empirical Legal Studies*, 9(1): 1–32, 2012.
- [134] Christopher D Manning, Prabhakar Raghavan, Hinrich Schütze, et al. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.
- [135] Alireza Mansouri, Lilly Suriani Affendey, and Ali Mamat. Named entity recognition approaches. *International Journal of Computer Science and Network Security*, 8(2):339–344, 2008.
- [136] Ryan T McDonald and Joakim Nivre. Characterizing the errors of data-driven dependency parsing models. In *EMNLP-CoNLL*, pages 122–131, 2007.
- [137] Edgar Meij, Wouter Weerkamp, and Maarten de Rijke. A query model based on normalized log-likelihood. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 1903–1906. ACM, 2009.
- [138] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
- [139] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.
- [140] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *hlt-Naacl*, volume 13, pages 746–751, 2013.

- [141] George A Miller and Walter G Charles. Contextual correlates of semantic similarity. *Language and cognitive processes*, 6(1):1–28, 1991.
- [142] David Milne and Ian H Witten. Learning to link with wikipedia. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 509–518. ACM, 2008.
- [143] Einat Minkov, Richard C Wang, and William W Cohen. Extracting personal names from email: Applying named entity recognition to informal text. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 443–450. Association for Computational Linguistics, 2005.
- [144] Muhidin Mohamed and Mourad Oussalah. Identifying and extracting named entities from wikipedia database using entity infoboxes. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 5(7), 2014.
- [145] Andrea Moro and Roberto Navigli. Semeval-2015 task 13: Multilingual all-words sense disambiguation and entity linking. *Proceedings of SemEval-2015*.
- [146] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1):3–26, 2007.
- [147] Joel Nothman, James R Curran, and Tara Murphy. Transforming wikipedia into named entity training data. In *Proceedings of the Australian Language Technology Workshop*, pages 124–132, 2008.
- [148] NTCIR. NTCIR Test Collections, 2015. URL <http://research.nii.ac.jp/ntcir/permission/data-en.htm>. [http://research.nii.ac.jp/ntcir/permission/data-en.htm; accessed 30-April-2016].
- [149] Mark K Osbeck. Using data analytics tools to supplement traditional research and analysis in forecasting case outcomes. *U of Michigan Public Law Research Paper Series*, (446), 2015.
- [150] Mark Osborn, Tomek Strzalkowski, and Mihnea Marinescu. Evaluating document retrieval in patent database: a preliminary report. In *Proceedings of the sixth international conference on Information and knowledge management*, pages 216–221. ACM, 1997.
- [151] Christos H Papadimitriou and Kenneth Steiglitz. *Combinatorial optimization: algorithms and complexity*. Courier Corporation, 1982.
- [152] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

- [153] Haoruo Peng, Yangqiu Song, and Dan Roth. Event detection and co-reference with minimal supervision. EMNLP, 2016.
- [154] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. *Proceedings of the Empirical Methods in Natural Language Processing (EMNLP 2014)*, 12:1532–1543, 2014.
- [155] Mohammad Taher Pilehvar and Roberto Navigli. From senses to texts: An all-in-one graph-based approach for measuring semantic similarity. *Artificial Intelligence*, 228:95–128, 2015.
- [156] Florina Piroi, Mihai Lupu, Allan Hanbury, Alan P Sexton, Walid Magdy, and Igor V Filippov. Clef-ip 2010: Retrieval experiments in the intellectual property domain. In *CLEF (notebook papers/labs/workshops)*, 2010.
- [157] Florina Piroi, Mihai Lupu, Allan Hanbury, and Veronika Zenz. Clef-ip 2011: Retrieval in the intellectual property domain. In *CLEF (notebook papers/labs/workshop)*. Citeseer, 2011.
- [158] Florina Piroi, Mihai Lupu, Allan Hanbury, Walid Magdy, Alan P. Sexton, and Igor Filippov. *CLEF-IP 2012: Retrieval experiments in the intellectual property domain*, volume 1178. CEUR-WS, 2012.
- [159] Florina Piroi, Mihai Lupu, and Allan Hanbury. *Information Access Evaluation. Multilinguality, Multimodality, and Visualization: 4th International Conference of the CLEF Initiative, CLEF 2013, Valencia, Spain, September 23-26, 2013. Proceedings*, chapter Overview of CLEF-IP 2013 Lab, pages 232–249. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. ISBN 978-3-642-40802-1. doi: 10.1007/978-3-642-40802-1_25. URL http://dx.doi.org/10.1007/978-3-642-40802-1_25.
- [160] Kira Radinsky, Eugene Agichtein, Evgeniy Gabrilovich, and Shaul Markovitch. A word at a time: computing word relatedness using temporal semantic analysis. In *Proceedings of the 20th international conference on World wide web*, pages 337–346. ACM, 2011.
- [161] K Rajshekhar, W Shalaby, and W Zadrozny. Analytics in post-grant patent review: Possibilities and challenges (preliminary report). 2016.
- [162] Kripa Rajshekhar, Walid Shalaby, and Wlodek Zadrozny. Analytics in post-grant patent review: Possibilities and challenges (preliminary report). In *Proceedings of the American Society for Engineering Management 2016 International Annual Conference*, 2016.
- [163] Lev Ratinov and Dan Roth. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pages 147–155. Association for Computational Linguistics, 2009.

- [164] Lev Ratinov and Dan Roth. Learning-based multi-sieve co-reference resolution with knowledge. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1234–1244. Association for Computational Linguistics, 2012.
- [165] Xiang Ren, Ahmed El-Kishky, Chi Wang, Fangbo Tao, Clare R Voss, and Jiawei Han. Clustype: Effective entity recognition and typing by relation phrase-based clustering. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 995–1004. ACM, 2015.
- [166] Alexander E Richman and Patrick Schone. Mining wiki resources for multilingual named entity recognition. In *ACL*, pages 1–9, 2008.
- [167] Petar Ristoski and Heiko Paulheim. Rdf2vec: Rdf graph embeddings for data mining. In *International Semantic Web Conference*, pages 498–514. Springer, 2016.
- [168] Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. Okapi at trec-3. *NIST SPECIAL PUBLICATION SP*, 109:109, 1995.
- [169] Joseph John Rocchio. Relevance feedback in information retrieval. *The SMART retrieval system: experiments in automatic document processing*, pages 313–323, 1971.
- [170] Giovanna Roda, John Tait, Florina Piroi, and Veronika Zenz. *Multilingual Information Access Evaluation I. Text Retrieval Experiments: 10th Workshop of the Cross-Language Evaluation Forum, CLEF 2009, Corfu, Greece, September 30 - October 2, 2009, Revised Selected Papers*, chapter CLEF-IP 2009: Retrieval Experiments in the Intellectual Property Domain, pages 385–409. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. ISBN 978-3-642-15754-7. doi: 10.1007/978-3-642-15754-7_47. URL http://dx.doi.org/10.1007/978-3-642-15754-7_47.
- [171] Herbert Rubenstein and John B. Goodenough. Contextual correlates of synonymy. *Commun. ACM*, 8(10):627–633, October 1965. ISSN 0001-0782. doi: 10.1145/365628.365657. URL <http://doi.acm.org/10.1145/365628.365657>.
- [172] Michail Salampasis and Allan Hanbury. Perfedpat: An integrated federated system for patent search. *World Patent Information*, 38:4–11, 2014.
- [173] Michail Salampasis, Anastasia Giachanou, and Allan Hanbury. An evaluation of an interactive federated patent search system. In *Multidisciplinary Information Retrieval*, pages 120–131. Springer, 2014.
- [174] G. Salton. *The SMART Retrieval System—Experiments in Automatic Document Processing*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1971.

- [175] Michael Schuhmacher and Simone Paolo Ponzetto. Knowledge-based graph document modeling. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 543–552. ACM, 2014.
- [176] David L Schwartz and Ted M Sichelman. Data sources on patents, copyrights, trademarks, and other intellectual property. *Copyrights, Trademarks, and Other Intellectual Property (August 17, 2015)*, 2, 2015.
- [177] Khaled Shaalan and Hafsa Raza. Person name entity recognition for arabic. In *Proceedings of the 2007 Workshop on Computational Approaches to Semitic Languages: Common Issues and Resources*, pages 17–24. Association for Computational Linguistics, 2007.
- [178] Walid Shalaby and Wlodek Zadrozny. Innovation analytics using mined semantic analysis. 2016.
- [179] Richard Socher, Jeffrey Pennington, Eric H Huang, Andrew Y Ng, and Christopher D Manning. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the conference on empirical methods in natural language processing*, pages 151–161. Association for Computational Linguistics, 2011.
- [180] Yangqiu Song and Dan Roth. Dataless Hierarchical Text Classification. https://cogcomp.org/page/software_view/Descartes, 2014. [Online; accessed 12-March-2018].
- [181] Yangqiu Song and Dan Roth. On dataless hierarchical text classification. In *AAAI*, pages 1579–1585, 2014.
- [182] Yangqiu Song and Dan Roth. Unsupervised sparse vector densification for short text similarity. In *Proceedings of NAACL*, 2015.
- [183] Yangqiu Song, Haixun Wang, Zhongyuan Wang, Hongsong Li, and Weizhu Chen. Short text conceptualization using a probabilistic knowledgebase. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Three*, pages 2330–2336. AAAI Press, 2011.
- [184] Yangqiu Song, Shusen Wang, and Haixun Wang. Open domain short text conceptualization: A generative+ descriptive modeling approach. In *IJCAI*, pages 3820–3826, 2015.
- [185] David Sontag and Dan Roy. Complexity of inference in latent dirichlet allocation. In *Advances in neural information processing systems*, pages 1008–1016, 2011.
- [186] Scott Spangler, Ying Chen, Jeffrey Kreulen, Stephen Boyer, Thomas Griffin, Alfredo Alba, Linda Kato, Ana Lelescu, and Su Yan. Simple: Interactive analytics on patent data. In *Data Mining Workshops (ICDMW), 2010 IEEE International Conference on*, pages 426–433. IEEE, 2010.

- [187] Scott Spangler, Chen Ying, Jeffrey Kreulen, Stephen Boyer, Thomas Griffin, Alfredo Alba, Linda Kato, Ana Lelescu, and Su Yan. Exploratory analytics on patent data sets using the simple platform. *World Patent Information*, 33(4): 328–339, 2011.
- [188] James H Steiger. Tests for comparing elements of a correlation matrix. *Psychological bulletin*, 87(2):245, 1980.
- [189] Tomek Strzalkowski. Natural language information retrieval. *Information Processing & Management*, 31(3):397–417, 1995.
- [190] Chade-Meng Tan, Yuan-Fang Wang, and Chan-Do Lee. The use of bigrams to enhance text categorization. *Information processing & management*, 38(4): 529–546, 2002.
- [191] Wolfgang Tannebaum and Andreas Rauber. Acquiring lexical knowledge from query logs for query expansion in patent searching. In *Semantic Computing (ICSC), 2012 IEEE Sixth International Conference on*, pages 336–338. IEEE, 2012.
- [192] Wolfgang Tannebaum and Andreas Rauber. Analyzing query logs of uspto examiners to identify useful query terms in patent documents for query expansion in patent searching: a preliminary study. In *Multidisciplinary Information Retrieval*, pages 127–136. Springer, 2012.
- [193] Wolfgang Tannebaum and Andreas Rauber. Mining query logs of uspto patent examiners. In *Information Access Evaluation. Multilinguality, Multimodality, and Visualization*, pages 136–142. Springer, 2013.
- [194] Wolfgang Tannebaum and Andreas Rauber. Using query logs of uspto patent examiners for automatic query expansion in patent searching. *Information Retrieval*, 17(5-6):452–470, 2014.
- [195] Wolfgang Tannebaum and Andreas Rauber. Patnet: a lexical database for the patent domain. In *Advances in Information Retrieval*, pages 550–555. Springer, 2015.
- [196] Wolfgang Tannebaum, Parvaz Mahdabi, and Andreas Rauber. Effect of log-based query term expansion on retrieval effectiveness in patent searching. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, pages 300–305. Springer, 2015.
- [197] Erik F Tjong Kim Sang and Fien De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 142–147. Association for Computational Linguistics, 2003.
- [198] Manuel Trajtenberg. A penny for your quotes: patent citations and the value of innovations. *The Rand Journal of Economics*, pages 172–187, 1990.

- [199] Peter D Turney, Patrick Pantel, et al. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37(1):141–188, 2010.
- [200] Suzan Verberne and Eva D’hondt. Prior art retrieval using the claims section as a bag of words. In *Multilingual Information Access Evaluation I. Text Retrieval Experiments*, pages 497–501. Springer, 2009.
- [201] Manisha Verma and Vasudeva Varma. Patent search using ipc classification vectors. In *Proceedings of the 4th workshop on Patent information retrieval*, pages 9–12. ACM, 2011.
- [202] Daniele Vitale, Paolo Ferragina, and Ugo Scaiella. Classification of short texts by deploying topical annotations. In *European Conference on Information Retrieval*, pages 376–387. Springer, 2012.
- [203] Ellen M Voorhees. Using wordnet for text retrieval. *Fellbaum (Fellbaum, 1998)*, pages 285–303, 1998.
- [204] Jakub Wajda and Wlodek Zadrozny. *Challenging Problems and Solutions in Intelligent Systems*, chapter Prior-Art Relevance Ranking Based on the Examiner’s Query Log Content, pages 323–333. Springer International Publishing, Cham, 2016. ISBN 978-3-319-30165-5. doi: 10.1007/978-3-319-30165-5_15. URL http://dx.doi.org/10.1007/978-3-319-30165-5_15.
- [205] Metti Zakaria Wanagiri and Mirna Adriani. Prior art retrieval using various patent document fields contents. In *CLEF (Notebook Papers/LABs/Workshops)*, 2010.
- [206] Feng Wang and Lanfen Lin. Query construction based on concept importance for effective patent retrieval. In *Fuzzy Systems and Knowledge Discovery (FSKD), 2015 12th International Conference on*, pages 1455–1459. IEEE, 2015.
- [207] Shuting Wang, Zhen Lei, and Wang-Chien Lee. Exploring legal patent citations for patent valuation. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 1379–1388. ACM, 2014.
- [208] Yushi Wang, Jonathan Berant, Percy Liang, et al. Building a semantic parser overnight. In *ACL (1)*, pages 1332–1342, 2015.
- [209] Zhongyuan Wang and Haixun Wang. Understanding short texts. In *the Association for Computational Linguistics (ACL) (Tutorial)*, August 2016.
- [210] Zhongyuan Wang, Haixun Wang, and Zhirui Hu. Head, modifier, and constraint detection in short texts. In *Data Engineering (ICDE), 2014 IEEE 30th International Conference on*, pages 280–291. IEEE, 2014.

- [211] Ian Witten and David Milne. An effective, low-cost measure of semantic relatedness obtained from wikipedia links. In *Proceeding of AAAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy*, AAAI Press, Chicago, USA, pages 25–30, 2008.
- [212] Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Q Zhu. Probbase: A probabilistic taxonomy for text understanding. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pages 481–492. ACM, 2012.
- [213] Xiaobing Xue and W Bruce Croft. Transforming patents into prior-art queries. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 808–809. ACM, 2009.
- [214] Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. Joint learning of the embedding of words and entities for named entity disambiguation. *arXiv preprint arXiv:1601.01343*, 2016.
- [215] Yiming Yang and Jan O Pedersen. A comparative study on feature selection in text categorization. In *Icml*, volume 97, pages 412–420, 1997.
- [216] Xiaoxin Yin and Sarthak Shah. Building taxonomy of web search intents for name entity queries. In *Proceedings of the 19th international conference on World wide web*, pages 1001–1010. ACM, 2010.
- [217] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*, 2014.
- [218] Torsten Zesch, Christof Müller, and Iryna Gurevych. Using wiktionary for computing semantic relatedness. In *AAAI*, volume 8, pages 861–866, 2008.
- [219] Luke S Zettlemoyer and Michael Collins. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. *arXiv preprint arXiv:1207.1420*, 2012.
- [220] Longhui Zhang, Lei Li, Chao Shen, and Tao Li. Patentcom: A comparative view of patent document retrieval. 2015.
- [221] GuoDong Zhou and Jian Su. Named entity recognition using an hmm-based chunk tagger. In *proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 473–480. Association for Computational Linguistics, 2002.
- [222] D. Zwillinger and S. Kokoska. *CRC standard probability and statistics tables and formulae*. CRC, 1999.

APPENDIX A: LIST OF PUBLICATIONS

The contributions of this thesis has resulted in 8 publications in top and highly ranked venues and 4 under review submissions (listed below). In addition, a US patent grant was issued on the applications of MSA described in Chapter 4.

1. Walid Shalaby, Wlodek Zadrozny, and Sean Gallagher. "Knowledge based dimensionality reduction for technical text mining." In *Big Data (Big Data), 2014 IEEE International Conference on*, pp. 39-44. IEEE, 2014.
2. Walid Shalaby, Khalifeh Al Jadda, Mohammed Korayem, and Trey Grainger. "Entity Type Recognition using an Ensemble of Distributional Semantic Models to Enhance Query Understanding." In *Computer Software and Applications Conference (COMPSAC), 2016 IEEE 40th Annual*, vol. 1, pp. 631-636. IEEE, 2016.
3. Walid Shalaby and Wlodek Zadrozny. "Mined Semantic Analysis: A New Concept Space Model for Semantic Representation of Textual Data." In *2017 IEEE International Conference on Big Data (Big Data), Boston, MA, USA, 2017*, pp. 2122-2131 (doi: 10.1109/BigData.2017.8258160).
4. Walid Shalaby and Wlodek Zadrozny. "Semantic Representation using Explicit Concept Space Models". in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-2017)*.
5. Walid Shalaby and Wlodek Zadrozny. "SustaInno: Toward a Searchable Repository of Sustainability Innovations." In *Workshops at the Twenty-Ninth AAAI*

Conference on Artificial Intelligence. 2015.

6. Walid Shalaby and Wlodek Zadrozny. "Innovation Analytics Using Mined Semantic Analysis." In *FLAIRS Conference*, pp. 597-601. 2016.
7. Walid Shalaby, Kripa Rajshekhar, and Wlodek Zadrozny. "A Visual Semantic Framework for Innovation Analytics." In *AAAI*, pp. 4389-4390. 2016.
8. Walid A. Shalaby, Wlodek W. Zadrozny, and Kripa Rajshekhar. "Natural Language Relatedness Tool using Mined Semantic Analysis." *U.S. Patent No. 9,880,999. 30 Jan. 2018.*
9. Kripa Rajshekhar, Walid Shalaby, and Wlodek Zadrozny. "Analytics In Post-Grant Patent Review: Possibilities And Challenges". Proceedings of the International Annual Conference of the American Society for Engineering Management., American Society for Engineering Management (ASEM), 2016.
10. Walid Shalaby and Wlodek Zadrozny. "Patent Retrieval: A Literature Review." *arXiv preprint arXiv:1701.00324 (2017) (under review).*
11. Walid Shalaby and Wlodek Zadrozny. "Learning Concept Embeddings for Efficient Bag-of-Concepts Densification." *arXiv preprint arXiv:1702.03342 (2017) (under review).*
12. Walid Shalaby, Wlodek Zadrozny, and Hongxia Jin. "Beyond Word Embeddings: Learning Entity and Concept Representations from Large Scale Knowledge Bases." *arXiv preprint arXiv:1801.00388 (2018) (under review).*

13. Walid Shalaby and Wlodek Zadrozny. "Toward an Interactive Patent Retrieval Framework based on Distributed Representations." (under review).