Pattern-based methods for Improved Lexical Semantics and Word Embeddings

Thesis submitted for the degree of
"Doctor of Philosophy"

By
Roy Schwartz

Submitted to the Senate of the Hebrew University
July, 2016

This work was carried out under the supervision of:
Prof. Ari Rappoport

Acknowledgments

Growing up as the youngest child in a family of five, I always felt I had the privilege of being raised by four parents. The love and care I received from my parents, Zipi and Yosi, and my brothers, Ori and Amir, as well as the different set of views and perspectives on life, made me who I am today, and words cannot express my deep and profound gratitude for that. Luckily, I was able to apply a similar model to my Ph.D., where other than (one) wonderful advisor, I also had several older lab members who served as my (mini-)advisors, and taught me much of what I know about research, academy and science. As a paraphrase to one of my favorite quotes goes: "*I do not know of a better way for a man to do a Ph.D.*".

Today another chapter of my journey is ending. This journey started about 9 years ago, when I met Prof. Ari Rappoport after a fascinating seminar talk about language, culture and the brain. Not long after, Ari became my advisor, and continued to play that role during my "Amirim" undergrad project, my master's and my Ph.D. Ari is undoubtedly one of the smartest people I know. His speed of processing and perception of complex ideas, and his ability to say the right thing in just a few words, keeps amazing me to this day. I think the most common scene I will take from my relationship with Ari is him suggesting an idea I do not understand at the time, and long afterwards, I suddenly realize what he was talking about all along. This has happened to me more than I can remember (though I am happy to say that with decreasing frequency).

I was privileged to work with Dr. Omri Abend, who guided me through my master's thesis. Omri was the one that taught me how to do research: how to think of a research question, how to address it, and how to write it into a scientific paper. It was only natural to witness him fulfill his dream and become a faculty member at the Hebrew University, a position that fits very few people better than him.

I owe a sincere gratitude to Dr. Roi Reichart. Roi was the one that encouraged me to pursue my very first scientific discovery, one that would develop into my master's thesis. A few years later, Roi joined the Technion as a faculty member, and we resumed our collaboration. Roi has served as a guide, a partner, and a friend during the last few years. He has taught me much about research, and how a simple finding could turn into a great paper. He is an endless ocean of ideas and useful insights, and is never satisfied by anything less than perfect. I am confident our paths will cross again.

I was only privileged to work with the late Dr. Dmitry Davidov when we co-taught the object-oriented programming course. His tragic death was a shock to us all, and a great loss to the NLP community. As it turned out, my Ph.D. largely relied on the strong foundation he has established. I only hope that my work is able to meet the high standards he has set.

I was very fortunate to meet and collaborate with several remarkable people, which I wish to deeply thank. Dr. Oren Tsur has a very unique view on science, and an eye for the most interesting problems. Dana Rubinstein is smart, fast learning, and fun to work with. Prof. Amir Globerson and Prof. Moshe Koppel always shared a piece of helpful advice. Dr. Ofer Meshi is very insightful, and I have learned a lot from him.

A few doors were always opened for questions, comments and advice. Prof. Avinoam Kolodny's help, advice (both technical and strategic), and good spirit were priceless. I would also like to deeply thank Dr. Aviv Zohar, who also I had the honor to have on my Ph.D. advisory board, Prof. Yair Weiss and Dr. Dafna Shahaf for their time, efforts and smiles.

I got a chance to share a lab with a wonderful group of people, who have made the last five years a memorable and fun time. I would like to warmly thank Amit Beka, Sagi Herman, Daniel Hershcovich, Eva Kimel, Effi Levi, Adi Littman and Elior Sulem. I would also like to thank a few precious people from across the hall, who created a rich social and academic environment: Cobi Cario, Haim Dubossarsky, Dr. Elad Mezuman, Dan Rosenbaum, Nir Rosenfeld, Dr. Or Sattath, Assaf Spanier, Ofer Springer, Rafi Vivanti and Dr. Daniel Zoran.

One special person who has played an important role in my long journey is Dr. Elad Eban. Throughout my master's and most of my Ph.D., Elad was a person to talk to about science, family, life, food and and whatever matters. He was and still is a very close friend, and our friendship is one of the most valuable things I have earned in this period.

I would like to thank the Hoffman program for leadership and responsibility for creating a unique and incredible opportunity to do something different, and make a real contribution to people's life. I specifically want to thank the Hoffman family – Harry, Sylvia, Sue and Josh, for their kind support and great spirit. I want to thank the program organizers, Prof. Amalya Oliver-Lumerman, Prof. Roi Baer, Prof. Hanoch Gutfreund, and of course Lital Myers, for making this dream a reality. And last, I want to thank the Hoffman program participants, the Fifth cohort in particular, for countless memorable moments.

Teaching was a very dominant part of my Ph.D. I would like to express my deep appreciation and gratitude to my advisor, Prof. Ari Rappoport, Prof. Dani Lischinski, Prof. Daphna Weinshall, Prof. Orna Kupferman, and the school of computer science and engineering in general, for putting their trust on a young graduate student, in teaching a course which at its peak reached 500 students and 15 TAs. I was also privileged to co-teach and work

with several wonderful people throughout the years. I want to deeply thank Nir Ben-Zvi, Noa Cohen, Inbar Huberman, Dan Nirel, and all the other TAs and exercise graders, who have made this an enjoyable and fulfilling time.

I want to thank the school of computer science and engineering administrative team, who were always extremely helpful, professional, and not less importantly – kind: Adel Aharon, Yael Bar David, Silvia Belisha, Yael Ben-Lulu, Ayelet Cohen, Rina Cohen, Miriam Edery, Zohar Marcovich, Hagit Yaar-On and Jacob Yagen.

A few friends were there to celebrate the joys and provide support in the less joyful moments. I want to deeply thank Dekel Avrahami, Uri Ayalon, Dr. Uri Ben-David, Ravid Ben-Zeev, Roee Cohen, Yehuda Pilnyak, Mark Polak, Yoav Wilf, Ronen Yom-Tov and Yotam Zeira, as well as the pricesless members of the "Halom" community, for the beers, the meals, the laughs and the good time.

Last, but definitely not least, I want to thank the people who have made it all worth it. The reason to get up in the morning, and the reason to come home after work: my beloved wife Tamar, and my two adorable sons, Yoav and Gilad.

Abstract

Natural Language Processing (NLP) is a field of research that aims, on the one hand, to give computational answers to linguistic questions, and on the other hand, develop applications for language oriented tasks, such as machine translation, text summarization and question answering. These two goals share a fundamental question, namely how to represent *semantics*; e.g., what is the meaning of the word *dog*, of expressions such as *Red Herring* or *kick the bucket*, or even more complex language structures. This question is linguistic (or even cognitive) in nature, while having direct empirical implications. Inside semantics, one of the most important subfields is *lexical* semantics, which studies the meaning or language utterances such as words or subwords.

The most prominent method for representing the semantics of language utterances is constructing feature vectors. These methods, also called *vector space models*, date back to the early 1970s. Until recent years, vector space models built co-occurrence matrices, such that each word is directly represented by the other words with which it co-occurs. In the last few years, novel approaches, often referred to as *word embeddings*, were developed for this task. These models were by large based on neural network algorithms, and were able to obtain substantial improvements on various semantic tasks. This success has made word embeddings a very popular tool, and created a sense that these models represent the complete semantics of a given word.

In this dissertation, I will show that despite their tremendous success, word embeddings suffer from several limitations. First, I will show that while state-of-the-art word embeddings are exceptionally well at capturing word *association* (e.g., “cup” is associated with “coffee”), they are far worse at capturing word *similarity* (e.g., “cup” is similar to “glass”). Second, I will demonstrate that they cannot distinguish between *similar* (“good”/“great”) and *opposite* words (“good”/“bad”). Third, I will show that while word embeddings are successful at capturing the semantics of *nouns* (e.g., “house”, “dog”), they are far less successful in capturing *verb* semantics (e.g., “run”, “walk”).

In order to address these problems, I will present a set of *pattern*-based solutions. Lexico-syntactic patterns (e.g., “X such as Y”, “X is a Y”) are one of the most effective alternatives to word embeddings in the task of semantic representation. They have been shown useful for capturing a wide range of semantic relations, including synonymy, hyponymy (is-a), antonymy (opposite-of), and many other relations. I will show that integrating patterns into word embeddings can greatly alleviate their problems.

I will start by demonstrating that pattern based methods can be superior to word embeddings on semantic tasks. I will present a variant of the k-Nearest Neighbors algorithm, which uses *symmetric* patterns (e.g., “X and Y”, “X or Y”) to capture a range of semantic properties (e.g., animacy, edibility, etc.), and obtains substantial improvements over state-of-the-art embeddings.

I will continue by presenting two word embedding models that are based on symmetric patterns, and are able to overcome the limitations of word embeddings. The first is a count based model that substantially outperforms six state-of-the-art embeddings on a word similarity task. The second is a pattern-based variant of the omnipresent word2vec skipgram model (Mikolov et al., 2013b), which outperforms it by more than 15% on a verb similarity task, while training in only a fraction of the original skipgram model training time.

For completeness, I will present another pattern-based system, which shows that patterns serve as valuable features for other tasks as well. I will present a pattern-based authorship attribution system, which obtains state-of-the-art results on the task of recognizing the author of a single tweet.

To conclude, the contribution of this dissertation lies in two aspects. First, it sheds light on the limitations (and strengths) of state-of-the-art word embeddings, which have until recently been considered almost omnipotent. Second, it demonstrates the power of patterns in overcoming some of these limitations, both by integrating pattern features into existing models and by developing novel, pattern-based models.

The Student's Contribution

Chapters 2-4 are all joint works with Dr. Roi Reichart and my advisor, Prof. Ari Rappoport. In each of these chapters, my contribution was in establishing the research idea, implementing the system, performing the experiments, the evaluation and the analysis, and writing the paper. Dr. Reichart's contribution was in providing numerous pieces of advice throughout the project on different matters, including the design of the research question, the model and the experiments. Dr. Reichart also provided feedback on the writing process.

Chapter 5 is a joint work with Dr. Oren Tsur, my advisor, Prof. Ari Rappoport and Prof. Moshe Koppel. As in the previous chapters, I was responsible for the entire pipeline, including the research question, the code, the experiments, the evaluation, the analysis and the writing. Dr. Tsur and Prof. Koppel's contribution was in providing high level comments and advice throughout the project, and providing feedback during the writing process. Prof. Koppel also took part in designing the research question.

Prof. Rappoport took a major part in establishing the general framework of this entire dissertation. In each of the chapters, he provided high-level, design comments throughout the process, including the research idea, the experimental process and the writing.

Contents

Chapter 1

Introduction

Natural Language Processing (NLP) is a field of research that focuses on the automatic representation and understanding of text. Research in NLP aims, on the one hand, to improve linguistic and cognitive understanding of language, and on the other hand, to develop natural language applications such as machine translation, question answering and textual search.

Research in NLP can be very broadly divided into two major fields: syntax and semantics. *Syntactic* NLP tasks focus on the induction of the syntactic structure of natural language sentences, such as part-of-speech (PoS) labels (Toutanova et al., 2003; Abend et al., 2010; Christodoulopoulos et al., 2010), phrase structure trees (Collins, 2003; Klein and Manning, 2004; Seginer, 2007) and dependency trees (Nivre and Hall, 2005; McDonald et al., 2005; Kübler et al., 2009).

Semantic tasks focus on the *meaning* of language utterances, and offer a wider range of tasks, starting from shallow semantics such as distributional semantics (Harris, 1954), through sentence levels semantic tasks such as semantic role labeling (Palmer et al., 2010) and semantic parsing (Zelle and Mooney, 1996; Zettlemoyer and Collins, 2005), up to tasks that require higher level understanding of text, such as statistical machine translation (Koehn, 2009) and text summarization (Mani and Maybury, 1999). This dissertation will focus on one of the most dominant NLP subfields nowadays – lexical semantics.

Research in lexical semantics focuses on extracting semantic features of lexical units such as words or sub-words. This subfield offers a wide range of NLP tasks, including measuring the degree of association or similarity between words and extracting semantic relations between words (e.g., synonyms, antonyms). The two main approaches to lexical semantics in NLP are distributional models (a.k.a. *vector space models* or *word embeddings*) and *lexico-syntactic patterns*. In this dissertation I will show that despite their increased popularity in recent years, distributional models suffer from serious shortcomings. I will then show that patterns can provide a better solution to semantic tasks, and that a combination of the two approaches can overcome many of these shortcomings.

Vector space models represent the meaning of words as vectors of real numbers. These vectors are the result of an algorithmic operation on the co-occurrence counts of a word and its neighboring words in natural language corpora. The theoretical foundation of these models is the distributional semantics hypothesis (Harris, 1954), which states that words that occur in similar contexts tend to have similar meanings. An important note here is that the term *context* is often interpreted as "bag-of-words context", i.e., the contexts of a word are the words surrounding it, regardless of the syntactic and/or semantic relation between them. While the earliest vector space models date back to the early 1970's (Salton, 1971), this field has remained active for several decades.

In the last few years distributional models have become extremely popular, with the introduction of novel neural-network based distributional models (a.k.a. *word embeddings*), such as C&W (Collobert et al., 2011), word2vec (Mikolov et al., 2013b) and GloVe (Pennington et al., 2014). These models obtained superb results on many lexical semantic tasks, including word association, word analogy and synonym detection (Baroni et al., 2014).[1] To quote: "*It seems possible to us that all of the semantics of human language might one day be captured in some kind of Vector Space Model*" (Turney and Pantel, 2010).

Despite their tremendous success and popularity, word embeddings are not flawless. In this dissertation, I will shed light on some of their limitations. First, I will show that word embeddings do not capture word *similarity* ("car" is similar to "train"), but word *association* (e.g., "car" and "wheal" are associated, Chapter 3). Second, I will demonstrate that state-of-the-art word embeddings are unable to distinguish between *similar* words ("good" and "great") and *opposite* words ("good" and "bad", Chapter 3). Third, I will show that there is a large discrepancy between their ability to capture *noun* similarity (e.g., "tiger" and "leopard") and their ability to capture *verb* similarity ("walk" and "run", Chapters 3,4), which is substantially inferior.

I will present solutions to these problems. These solutions are based on lexico-syntactic patterns. Patterns are sequences of words and wildcards (e.g., "such X as Y"). Ever since their introduction in the early 1990's (Hearst, 1992), patterns have been shown useful for capturing various types of word relations, such as hypernymy ("cat"/"animal", Hearst (1992); Snow

[1] Although see (Levy et al., 2015a) for a disclaimer.

et al. (2004)), antonymy (“good”/“bad”, Lin et al. (2003)) and synonymy (Turney, 2008a). In this dissertation, I will show that patterns can be used to overcome many of the limitation of state-of-the-art word embeddings, either by developing new models (Chapter 2) or by integrating them in existing word embedding models (Chapters 3,4).

In Chapter 2 I will start by showing that patterns can serve as better features compared to leading word embeddings in semantic tasks. I will present a novel algorithm based on the k-Nearest-Neighbors algorithm, and show that applying it with symmetric patterns features leads to substantial improvements over state-of-the-art word embeddings on the task of minimally supervised word classification.

I will continue by presenting two word embedding models that integrate pattern contexts in order to overcome their limitations. In Chapter 3 I will present a vector space model that replaces bag-of-word contexts with *symmetric* patterns (“X and Y”). I will show that this model (a) captures word similarity and not association, (b) is able to distinguish between similar and opposite words, and (c) preforms exceptionally well on verb similarity.

In Chapter 4 I will show that integrating symmetric patterns into the omnipresent word2vec skipgram with negative sampling model (Mikolov et al., 2013b) improves its verb similarity performance by 15%, and also results in a model that is dramatically faster to train.

In Chapter 5 I will demonstrate that patterns are also useful in other domains, by presenting a pattern-based authorship attribution system for very short texts (tweets). My experiments show that by incorporating pattern features, this system is able to reach state-of-the-art performance.

Background

Lexical Semantics

Lexical Semantics is a subfield in linguistics that studies the meaning of lexical units (mostly words, but also multiword expressions or sub-words). The goal of this research is to characterize words in terms of their features (e.g., “dog” is an animate noun, “water” is a mass noun), and their relation to other words. Such relations include, among others, synonymy (“big” / “large”), antonymy (“good” / “bad”), hypernymy (“animal” / “dog”) and meronymy (“wheel” / “car”).

In NLP, several approaches to lexical semantics have been proposed. The first approach is constructing lexicons manually (or semi-automatically). The most notable example is WordNet (Miller, 1995), which is a large lexical resource for English which focuses on taxonomic relations. Other resources include EuroWordNet (Vossen, 1998), which extends WordNet to other European languages, and other lexicons that focus on verb relations, such as COMLEX (Grishman et al., 1994) and VerbNet (Schuler, 2005). Several ontologies, such as Yago (Suchanek et al., 2007) and BabelNet (Navigli and Ponzetto, 2012), were assembled by automatically harvesting resources such as WordNet and Wikipedia

Other approaches to lexical semantics include compiling vector representation of words (a.k.a. *vector space models* or more recently, *word embeddings*) and extracting *lexico-syntactic patterns*. Below is a short background to both approaches.

Vector Space Models

Vector space models are computational models for representing words as vectors of real numbers. Research on vector space models dates back to the early 1970's (Salton, 1971). Until recent years, vector space models represented each word w as a vector in which each coordinate represents the co-occurrence of w and another word w'. Importantly, the great majority of vector space models consider very little information on the syntactic and semantic relations between w and w'. Instead, a bag-of-words approach is taken. The resulting vectors are often post-processed by weighting techniques such as Positive Pointwise Mutual Information (PPMI) normalization and dimensionality reduction methods such as Singular Value Decomposition (SVD). For recent surveys, see (Turney and Pantel, 2010; Clark, 2012; Erk, 2012).

Evaluation of Vector Space Models There are several ways to evaluate the quality of vector space models. The most common approach is to generate pairs of words, and have humans annotate their degree of similarity or association. Several efforts have been made along this line over the years, including RG-65 (Rubenstein and Goodenough, 1965), MC-30 (Miller and Charles, 1991), WordSim353 (Finkelstein et al., 2001), MEN (Bruni et al., 2014) and SimLex999 (Hill et al., 2015).[2] The evaluation procedure includes comparing the human scores with the cosine similarity scores between each pair of vectors, and then computing the correlation (usually in terms of Spearman's ρ) between the relative rankings.

Other types of evaluation were also proposed. In the TOEFL task (Freitag et al., 2005), the system is presented with one target word and four possibles choices, one of which is its synonym. The evaluation procedure checks whether the word with the highest cosine similarity is indeed the syn-

[2] For a comprehensive list see: `wordvectors.org/`

onym. Other tasks include treating the vector elements as features for word classification or clustering tasks (Almuhareb, 2006; Baroni et al., 2008).

In addition to intrinsic evaluation, vector space models are also evaluated in extrinsic tasks (e.g., Named-Entity Recognition (Turian et al., 2010)), where they can either replace or be applied on top of lexical features. Moreover, many neural network models use intermediate vector representation, e.g., the Stanford Neural Network Dependency parser (Chen and Manning, 2014). Different vector space models can thus be evaluated in terms of how much they benefit a given system.

Interestingly, recent works have pointed out to several methodological problems in the evaluation setup of vector space models (Faruqui et al., 2016), such as a low correlation between extrinsic and intrinsic evaluation (Schnabel et al., 2015; Tsvetkov et al., 2015; Melamud et al., 2016). The large number of evaluation measures, as well as their inconsistency, make the field of vector space evaluation an open research question.

Word Embeddings Recently, a line of work presented Neural Network (NN) algorithms for vector space modeling (Bengio et al., 2003; Collobert and Weston, 2008; Mnih and Hinton, 2009; Collobert et al., 2011; Dhillon et al., 2011; Mikolov et al., 2013b; Mnih and Kavukcuoglu, 2013; Lebret and Collobert, 2014; Pennington et al., 2014). These models went by the name *word embeddings*. Like earlier works, these models also follow the bag-of-words approach. However, they encode this information into their objective, often a language model, rather than directly into the features.

Word embeddings have shown to be successful in various semantic tasks, such as word association, synonym detection and word clustering (Baroni et al., 2014). This trend of works also introduced a new evaluation task – word analogy test (Mikolov et al., 2013c). In this test, the embeddings are used to compute analogies such as "*man* is to *woman* like *king* is to ?" (*queen*). Mikolov et al. (2013c) presented the word2vec skip-gram with negative sampling model, and showed that the result of the vector computation of v_{king}-v_{man}+v_{woman} is a vector that is most similar to v_{queen}.

Other than capturing semantic information, word embeddings have also shown useful for downstream tasks such as Named-Entity recognition (Turian et al., 2010), semantic role labeling (Collobert et al., 2011), sentiment analysis (Socher et al., 2013), machine translation (Devlin et al., 2014) and dependency parsing (Chen and Manning, 2014). This empirical success has made word embeddings a very popular field of research in the last few years.

Model	Nouns	Verbs
GloVe	0.377	0.163
BoW	0.451	0.276
CBOW	0.48	0.252
skip-gram	0.501	0.307

Table 1.1:
Spearman's ρ performance of four leading word embedding models on noun and verb similarity tasks. Models: Glove (Pennington et al., 2014), BoW – bag-of-words co-occurrence model with PPMI weighting, CBOW – word2vec CBOW (Mikolov et al., 2013a), skip-gram – word2vec skipgram (Mikolov et al., 2013b). All models perform substantially better on nouns than on verbs.

Limitations of Word Embeddings Despite their tremendous success in many semantic tasks and downstream applications, word embeddings are not perfect. In Chapters 3,4 we highlight a few problems with state-of-the-art embeddings. First, while bag-of-words word embeddings capture word association to a very high degree, they fail to distinguish between *associated* and *similar* words. Consider the pair of words "cow" and "milk". These two words co-occur quite frequently, and thus words that co-occur with one of them, also tend to co-occur with the other. Consequently, a vector space model constructed with bag-of-words features will assign similar vectors to this pair of words. Indeed, the word2vec skip-gram embeddings (Mikolov et al., 2013b) cosine similarity score between these words is relatively high (0.61). Similarly, the similarity score assigned by this model to the word pair "computer" and "software" is 0.68.

Second, state-of-the-art word embeddings fail to distinguish between similar and opposite words. This is because much like similar words, opposite words (*antonyms*) also tend to occur in the same context, and thus their word embeddings are often similar. For example, the skip-gram score of the (*accept*,*reject*) pair is 0.73, and the score of (*long*,*short*) is 0.71.

Third, there is a large discrepancy between the performance of state-of-the art embeddings on *noun* related tasks and *verb* related tasks. Table 1.1 shows that performance of four leading bag-of-words embeddings on the noun and verb portions of the SimLex999 word similarity dataset (Hill et al., 2015). The table shows that the performance of all models decreases by 17.5%-22.6% when shifted from nouns to verbs.

Finally, a few recent papers examined the limitations of word embeddings in representing different types of semantic information. Levy et al.

(2015b) showed that word embeddings do not capture semantic relations such as hyponymy and entailment. In (Rubinstein et al., 2015), we showed that while state-of-the-art embeddings are successful at capturing taxonomic information (e.g., *apple* is a fruit), they are much less successful in capturing attributive properties (*elephants* are big).

Solutions to the Limitations of Word Embeddings A few works tackled some of the problems mentioned above by injecting lexical knowledge, such as dictionary, ontology or thesaurus information, into the embeddings. This knowledge can be injected into the objective function (Kiela et al., 2015; Pham et al., 2015; Liu et al., 2015), or as a post-processing step (Faruqui et al., 2015; Mrkšić et al., 2016). This external knowledge enabled the resulting embeddings to differentiate between related and similar pairs of words, and in some cases also between similar and opposite word pairs.

A few works tackled these problems in a corpus-based method, without using external knowledge resources. The main approach in these works is to replace bag-of-words contexts with other context types, which consider deeper relationships between linguistic items. A few works represented words through their co-occurrence with other words in syntactic dependency relations (Lin, 1998; Padó and Lapata, 2007; Murphy et al., 2012; Levy and Goldberg, 2014). Other works used other types of contexts, such as clusters of lexico-syntactic patterns (Baroni et al., 2010; Bollegala et al., 2015). Yatbaz et al. (2012) replaced bag-of-word contexts with substitute vectors, which include the potential words that could replace the target word given its neighboring words. The main benefit from these models is to build more functional embeddings, in order to capture similarity rather than relatedness (Levy and Goldberg, 2014).

In this dissertation, I present (Chapters 3,4) two word embedding models based on *symmetric* patterns, and one model based on syntactic coordination, and show that these models alleviate many of the problems discussed earlier. Similarly to the other models that present alternatives to bag-of-words contexts, these embedding models also capture word similarity rather than relatedness. Nonetheless, symmetric pattern contexts have several advantages compared to the solutions presented above. First, in contrast to using dependency-based contexts, symmetric patterns are computed in a fully unsupervised manner, and are thus applicable to any language. Second, the proposed methods result in a particularly accurate representation of verbs, as reflected by their state-of-the-art results on verb similarity – 20% improvement over the second best baseline (Chapter 3). Third, a symmetric-pattern based approach is dramatically faster to train. For instance, training the word2vec skip-gram model (Mikolov et al., 2013c) with symmetric pattern contexts takes only 2-3% of the time it takes to train the model on the same corpus with bag-of-words or dependency contexts. Finally, other than capturing similarity rather than relatedness, the method proposed in Chapter 3 is also able to distinguish between similar and opposite words. To the best of my knowledge, this is the only corpus based word embedding model that can make this distinction.

Lexico-Syntactic Patterns

A second approach to lexical semantics is lexico-syntactic patterns. Patterns are sequences of words and wildcards (Hearst, 1992). Examples of patterns include "X *such as* Y", "X *or* Y" and "X *is a* Y". When patterns are instantiated in text, wildcards are replaced by words. For example, the pattern "***X*** *is a* ***Y***", with the **X** and **Y** wildcards, can be instantiated in phrases like "***Goofy*** *is a* ***dog***".

Patterns were shown useful for capturing a wide range of semantic relations. Hearst (1992) used patterns like "X *such as* Y" and "*such* Y *as* X" to extract hypernym/hyponym relations (animal/dog). Berland and Charniak (1999) applied patterns such as "X of a Y" for the detection of the meronymy (part-of) relation (building/basement). Lin et al. (2003) used the patterns "*from* X *to* Y" and "*either* X *or* Y" to extract antonym relations (good/bad). *Symmetric* patterns (e.g., "X *and* Y") have been shown useful for capturing word similarity (Widdows and Dorow, 2002; Davidov and Rappoport, 2006).

Patterns were also used in general tasks such as knowledge extraction (Etzioni et al., 2005) and general word relations (Davidov and Rappoport, 2008a,b). In addition, they have successfully served as features for downstream tasks such as detection of sarcasm (Tsur et al., 2010), sentiment analysis (Davidov et al., 2010), minimally supervised word classification (Schwartz et al. (2014), Chapter 2) and authorship attribution (Schwartz et al. (2013), Chapter 5).

Symmetric Patterns *Symmetric* patterns are a special type of patterns that contain exactly two wildcards and that tend to be instantiated by wildcard pairs such that each member of the pair can take the **X** or the **Y** position. For example, the symmetry of the pattern "***X*** *or* ***Y***" is exemplified by the semantically plausible expressions "cats *or* dogs" and "dogs *or* cats". In contrast, "***X*** *is a* ***Y***" is *asymmetric* because pairs of words that co-occur in one position (e.g., "Rihanna *is a* singer"), do not tend to co-occur in the other position ("*singer *is a* Rihanna").

Previous works have shown that words that co-occur in symmetric patterns tend to be semantically similar (Widdows and Dorow, 2002; Davidov and Rappoport, 2006). In this dissertation I start (Chapter 2) by demonstrating that symmetric patterns can serve as better features than state-of-the-art embeddings in lexical semantics tasks. I present a weakly supervised model for semantic word classification. The model applies the iterative k-Nearest Neighbors (I-k-NN) algorithm, a novel variant of the k-Nearest-Neighbors algorithm, which is particularly suited for minimally supervised tasks. The novel model uses symmetric pattern counts as edge weights, and requires only a handful of training examples. In my experiments, with four different semantic categories (e.g., edible nouns, animate nouns), this model obtains an average improvement of 15% over two leading models. Moreover, the presented I-k-NN algorithm also performs better than two state-of-the-art minimally supervised classification algorithms.

I continue by showing that patterns can be integrated into word embeddings in order to overcome their limitations presented earlier. I present two models that use symmetric patterns to generate embeddings that capture word similarity rather than relatedness.

- The first work (Chapter 3) presents a co-occurrence model that replaces bag-of-word counts with symmetric pattern counts. As words that co-occur in symmetric patterns tend to be similar (and not simply related, as in bag-of-words co-occurrence), the resulting vectors are able to distinguish between similar and related pairs.

  That work also presents *negative weighting*, a novel antonym detection mechanism, which prevents the model from assigning similar vectors to opposite words. This mechanism follows the work of (Lin et al., 2003), who found that antonym pairs tend to co-occur in the *antonym* patterns "*from* X *to* Y" and "*either* X *or* Y". With this observation, I subtract the *antonym* pattern counts from the *symmetric* pattern counts, and thus the resulting model assigns similar vectors to similar words, and different vectors to opposite words.

  This model obtains state-of-the-art results on the SimLex999 word similarity dataset (Hill et al., 2015), improving over six strong baselines by 5.5%-16.7%. Interestingly, the model performs exceptionally well on the verb portion of SimLex999, obtaining 20.2%-40.5% improvement over these models.

- The second work (Chapter 4) studies the effect of the context type on the performance of word embeddings. In this chapter, I demonstrate that replacing bag-of-word contexts with symmetric pattern contexts in the omnipresent word2vec skipgram model (Mikolov et al., 2013b) leads to up to 15% gain on a verb similarity task and up to a 9% gain on adjective similarity. Moreover, replacing the contexts also results in a much more compact model, which trains in only 3% of the training time of the bag-of-words skip-gram model. Finally, I also show that the symmetric patterns skip-gram variant performs better (although by a smaller margin) than a model based on syntactic coordinations, which are extracted using a supervised dependency parser. These findings demonstrate the power of symmetric patterns in the context of verb and adjective similarity.

Authorship Attribution of Short Texts

Authorship attribution is a multi-class classification task, where a system is trained on (*document,author*) pairs. Traditionally, authorship attribution systems have mainly been evaluated against long texts such as theater plays (Mendenhall, 1887), essays (Yule, 1939; Mosteller and Wallace, 1964), biblical books (Mealand, 1995; Koppel et al., 2011a) and book chapters (Argamon et al., 2007; Koppel et al., 2007). In the last decade or so, authorship attribution works started to focus on web data such as emails (De Vel et al., 2001; Koppel and Schler, 2003; Abbasi and Chen, 2008), web forum messages (Abbasi and Chen, 2005; Solorio et al., 2011), blogs (Koppel et al., 2006, 2011b) and chat messages (Abbasi and Chen, 2008).

In recent years, a few works focused on very short text such as SMS messages (Mohan et al., 2010; Ishihara, 2011) and Twitter tweets (Frantzeskou et al., 2007; Silva et al., 2011; Boutwell, 2011; Mikros and Perifanos, 2013). In Chapter 5, I present a pattern-based authorship attribution system that is specifically useful for very short texts. The system obtains state-of-the-art results on the Twitter domain – a 6% improvement over previous methods. Moreover, unlike previous works, which were limited to 200 authors at most, in this work I also present experiments with a very large pool of authors (up to 1,000 authors), showing that the system is able to reach reasonable performance in this setup (more than 30% accuracy). Finally, I introduce the concept of an author's unique "signature", and show that such signatures are typical of many authors when writing very short texts.

Summary of Research Achievements

The goal of this dissertation is to examine different ways of extracting lexical semantic knowledge. While word embeddings are considered a very effective tool for this task, this dissertation presented several, rather basic aspects of lexical semantics that they are currently unable to capture. These limitations question the assumption that these models capture all (or even most) of the semantic properties of words, and indicate that there is much room for improving these tools. The second part of this dissertation showed that lexico-syntactic patterns may serve as better features for lexical semantic tasks compared to word embeddings. It also showed that integrating patterns into existing embeddings alleviates many of the limitations of bag-of-words embeddings. This dissertation is just a first step in addressing the wide range of lexical semantic tasks. There is much room to investigate how patterns, embeddings or their combination can improve performance on other tasks, such as extraction of semantic relations.

Published in COLING 2014

Chapter 2

Minimally Supervised Classification to Semantic Categories using Automatically Acquired Symmetric Patterns

Minimally Supervised Classification to Semantic Categories using Automatically Acquired Symmetric Patterns

Roy Schwartz[1] **Roi Reichart**[2] **Ari Rappoport**[1]

[1]Institute of Computer Science, The Hebrew University
{roys02|arir}@cs.huji.ac.il

[2]Technion IIT
roiri@ie.technion.ac.il

Abstract

Classifying nouns into semantic categories (e.g., animals, food) is an important line of research in both cognitive science and natural language processing. We present a minimally supervised model for noun classification, which uses symmetric patterns (e.g., “X and Y”) and an iterative variant of the k-Nearest Neighbors algorithm. Unlike most previous works, we do not use a predefined set of symmetric patterns, but extract them automatically from plain text, in an unsupervised manner. We experiment with four semantic categories and show that symmetric patterns constitute much better classification features compared to leading word embedding methods. We further demonstrate that our simple k-Nearest Neighbors algorithm outperforms two state-of-the-art label propagation alternatives for this task. In experiments, our model obtains 82%-94% accuracy using as few as four labeled examples per category, emphasizing the effectiveness of simple search and representation techniques for this task.

1 Introduction

The role of language is to express meaning. In the field of NLP, there has been an increasingly growing number of approaches that deal with semantics. Among these are vector space models (Turney and Pantel, 2010; Baroni and Lenci, 2010), lexical acquisition (Hearst, 1992; Dorow et al., 2005; Davidov and Rappoport, 2006), universal cognitive conceptual annotation (Abend and Rappoport, 2013) and automatic induction of feature representations (Collobert et al., 2011). In this paper, we utilize extremely weak supervision to classify words into fundamental cognitive semantic categories.

There are several types of semantic categories expressed by languages, e.g., objects, actions, and properties. We follow human development, acquiring coarse-grained categories and distinctions before detailed ones (Mandler, 2004). Specifically, we focus on the major class of concrete “*things*” (Langacker, 2008, Ch. 4), roughly corresponding to nouns – the main participants in linguistic clauses – that are universally present in the semantics of virtually all languages (Dixon, 2005).

Most works on noun classification to semantic categories require large amounts of human annotation to build training corpora for supervised algorithms (Bowman and Chopra, 2012; Moore et al., 2013) or rely on language-specific resources such as WordNet (Evans and Orăsan, 2000; Orăsan and Evans, 2007). Such heavy supervision is labor intensive and makes these models domain and language dependent.

Our reasoning is that weak supervision is highly valuable for semantic categorization, as it can compensate for the lack of input from the senses in text corpora. Our model therefore performs semantic category classification using only a small number of labeled seed words per category. The experiments we conduct show that such weak supervision is sufficient to construct a high quality classifier.

A key component of our model is the application of symmetric patterns. We define patterns to be sequences of words and wildcards (e.g., “X is a dog”, “both X and Y”, etc.). Accordingly, *symmetric* patterns are patterns that contain exactly two wildcards, where both wildcards are interchangeable. Examples of symmetric patterns include “X and Y”, “X as well as Y” and “neither X nor Y”.

Works that apply symmetric patterns in their model generally require expert knowledge in the form of a pre-compiled set of patterns (Widdows and Dorow, 2002; Kozareva et al., 2008). In this work, we extract symmetric patterns in an unsupervised manner using the (Davidov and Rappoport, 2006) algorithm. This algorithm automatically extracts a set of symmetric patterns from plain text using simple statistics about high and low frequency word co-occurrences. The unsupervised nature of our approach makes it domain and language independent.

Our model addresses semantic classification in a transductive setup. It takes advantage of word similarity scores that are computed based on symmetric pattern features, and propagates information from concepts with known classes to the rest of the concepts. For this aim we apply an iterative variant of the k-Nearest Neighbors algorithm (denoted with I-k-NN) to a graph in which vertices correspond to nouns and word pairs are connected with edges based on their participation in symmetric patterns.

We experiment with a subset of 450 nouns from the CSLB dataset (Devereux et al., 2013), which were annotated with semantic categories by thirty human subjects. From the set of semantic categories in this dataset, we select categories that are both frequent and have a high inter-annotator agreement (Section 2). This results in a set of four semantic categories – *animacy, edibility, is_a_tool* and *is_worn*.

Our experiments show that our model performs very well even when only a small number of labeled seed words are available. For example, on the task of binary classification with respect to a single category, when using as few as four labeled seed words, classification accuracy reaches 82%-94%.

Furthermore, our model outperforms several strong baselines for this task. First, we compare our model against a model that uses a deep neural network word embedding baseline (Collobert et al., 2011) instead of our symmetric pattern based features, and applies the exact same I-k-NN algorithm. In recent years, deep networks word embeddings obtained state-of-the-art results in several NLP tasks (Collobert and Weston, 2008; Socher et al., 2013). However, in our task, features based on simple, intuitive and easy to compute symmetric patterns, lead to substantially better performance (average improvement of 0.15 F1 points). Second, our model outperforms two baseline models that utilize the same symmetric pattern classification features as in our model, but replace our simple I-k-NN algorithm with two leading label propagation alternatives (the normalized graph cut (N-Cut) algorithm (Yu and Shi, 2003) and the Modified Adsorption (MAD) algorithm (Talukdar and Crammer, 2009)). The average improvement over these two baselines is 0.21 and 0.03 F1 points .

The rest of the paper is organized as follows. Section 2 describes our semantic classification task and, particularly, the semantic classes that we aim to learn. Section 3 presents our method for automatic symmetric patterns acquisition. Sections 4, 5 and 6 describe our model, experimental setup and results, respectively. Related work is finally surveyed in Section 7.

2 Task Definition

The task we tackle in this paper is the classification of nouns into semantic categories. This section defines the categories we address and the dataset we use.

Semantic Categorization of Concrete Nouns. We focus on concrete "*things*" (Langacker, 2008), which correspond to *noun* categories. Nouns are interesting because they are the most basic lexical semantic categories. Specifically, children acquire nouns before any other category (Clark, 2009). Moreover, noun categories are generally not subjective. For example, it is hard to argue that a dog is not an animal, or that an apple is inedible, in most reasonable contexts. The context independent nature of nouns makes them appropriate for a type level classification task, such as the one we tackle. In order to provide a better description of the categories we aim to predict, we now turn to discuss the CSLB dataset, with which we experiment.

Dataset. We experiment with the CSLB property norms dataset (Devereux et al., 2013). In order to prepare this data set, thirty human subjects were presented with 638 concrete nouns and were asked to write the categories associated with each concept. Table 1 presents the top five categories for the nouns *apple* and *horse*.

Noun	Categories
Apple	is_a_fruit, does_grow_on_trees, is_green, is_red, has_pips_seeds
Horse	is_ridden, is_an_animal, has_four_legs, has_legs, has_hooves

Table 1: Five most frequent semantic categories for the words *apple* and *horse* in the CSLB dataset.

Category Selection. The CSLB dataset consists of a total of 2725 semantic categories. We apply a selection mechanism that provides us with a dataset in which (1) only noun categories (*things*) are included; and (2) only semantic categories that are prominent across humans are considered. For this, we apply the following filtering stages. First, since the vast majority of annotated categories are rare (for example, 1691 categories are assigned to a single noun only), we set a minimum threshold of 35 nouns per category (5% of the nouns). After removing highly infrequent categories, 28 are left. We then apply an inter-annotator agreement criterion: for each semantic category c, we compute the average number of human annotators that associated this category with a given noun, across the nouns annotated with c. We select the category c only if the value of this statistic is higher than 10 subjects ($1/3$ of the subjects), which results in a semantic category set of size 18. Finally, we discard categories, such as *color* and *size*, that do not correspond to *things*. We are left with four noun semantic categories: *animacy* (animals), *edibility* (food items), *is_a_tool* (tools), and *is_worn* (clothes).

Interestingly, the resulting semantic categories can also be justified from a cognitive perspective. There is a large body of work indicating that our categories relate to brain organization principles. For example, Just et al. (2010) showed that food products and tools arouse different brain activation patterns. Moreover, a number of works showed that both animate objects and tools are represented in specific brain regions. These works used neuroimaging methods such as functional magnetic resonance imaging (fMRI) (Naselaris et al., 2012), electroencephalography (EEG) (Chan et al., 2011) and magnetoencephalography (MEG) (Sudre et al., 2012). See (Martin, 2007) for a detailed survey. This parallel evidence to the prominence of our categories provides substance for intriguing future research.

3 Symmetric Patterns

Patterns. In this work, patterns are combinations of words and wildcards, which provide a structural phrase representation. Examples of patterns include "*X and Y*", "*X such as Y*", "*X is a country*", etc. Patterns can be used to extract various relations between words. For example, patterns such as "X of a Y" ("basement *of a* building") can be useful for detecting the meronymy (part-of) relation (Berland and Charniak, 1999). Symmetric patterns (e.g., "X *and* Y", "France *and* Holland"), which we use in this paper, can be used to detect semantic similarity between words (Widdows and Dorow, 2002).

Symmetric Patterns. *Symmetric* patterns are patterns that contain exactly two wildcards, and where these wildcards are interchangeable. Examples of symmetric patterns include "X *and* Y", "X *or* Y" and "X *as well as* Y". Previous works have shown that word pairs that participate in symmetric patterns bare strong semantic resemblance, and consequently, that these patterns can be used to cluster words into semantic categories, where a high precision, but low coverage (recall) solution is good enough (Dorow et al., 2005; Davidov and Rappoport, 2006). A key observation of this paper is that symmetric patterns can be also used for semantic classification, where recall is as important as precision.

Flexible Patterns. It has been shown in previous work (Davidov and Rappoport, 2006; Turney, 2008; Tsur et al., 2010; Schwartz et al., 2013) that patterns can be extracted from plain text in a fully unsupervised manner. The key idea that makes this procedure possible is the concept of "flexible patterns", which are composed of high frequency words (HFW) and content words (CW). Every word in the language is defined as either HFW or CW, based on the number of times this word appears in a large corpus. This clustering procedure is applied by traversing a large corpus, and marking words that appear with corpus frequency higher than a predefined threshold t_1 as HFWs, and words with corpus frequency lower than t_2 as CWs.[1]

[1] We follow (Davidov and Rappoport, 2006) and set $t_1 = 10^{-5}, t_2 = 10^{-3}$. Note that some words are marked both as HFW and as CW. See (Davidov and Rappoport, 2008) for discussion.

The resulting clusters have a desired property: HFWs are comprised mostly of function words (prepositions, determiners, etc.) while CWs are comprised mostly of content words (nouns, verbs, adjectives and adverbs). This coarse grained clustering is useful for pattern extraction from plain text, since language patterns tend to use fixed function words, while content words change from one instance of the pattern to another (Davidov and Rappoport, 2006).

Flexible patterns are extracted by traversing a large corpus and, based on the clustering of words to CWs and HFWs, extracting all pattern instances. An extracted pattern instance consists of CW wildcards and the actual words replacing the HFWs in the pattern type. Consider the sentence "*The boy is happy and joyful*". Replacing the content words with the CW wildcard results in "*The* CW *is* CW *and* CW". From this intermediate representation, we extract word sequences of a given length constraint and denote them as flexible patterns.[2] The flexible patterns of length 5 extracted from this sentence are "*The* CW *is* CW *and*" and "CW *is* CW *and* CW". The reader is referred to (Davidov and Rappoport, 2006) for more details.

Automatically Extracted Symmetric Patterns. Most models that incorporate symmetric patterns use a predefined set of patterns (Widdows and Dorow, 2002; Kozareva et al., 2008). In this work, we apply an automatic, completely unsupervised procedure for symmetric pattern extraction. This procedure, described in Algorithm 1, is adopted from (Davidov and Rappoport, 2006).

The procedure first extracts flexible patterns that contain exactly two CW wildcards. It then selects those flexible patterns in which both CWs are interchangeable. That is, it selects a pattern p if every word pair CW_1, CW_2 that participates in p indicates with high probability that the word pair C_2, C_1 also participates in p. For example, for the symmetric pattern "CW *and* CW", both "cats *and* dogs" and "dogs *and* cats" are semantically plausible expressions, and are therefore likely to appear in a large corpus. On the other hand, the flexible pattern "CW *such as* CW" is asymmetric, as exemplified in expressions like "countries *such as* France", where replacing the CWs does not result in a semantically plausible expression ($\#$ "France *such as* countries"). The selection process is done by computing the proportion of CW_1, CW_2 pairs that participate in p for which CW_2, CW_1 also participates in p. Patterns for which this proportion exceed a certain threshold are selected.

We apply Algorithm 1 on the google books 5-gram corpus (Michel et al., 2011)[3] and extract 20 symmetric patterns. Some of the more interesting symmetric patterns extracted using this algorithm include "CW *and the* CW", "*from* CW *to* CW", "CW *rather than* CW" and "CW *versus* CW". In the next section we present our approach to semantic classification, which makes use of automatically acquired symmetric patterns for word similarity computations.

4 Model

In this section we present our model for binary word classification according to a single semantic category in a minimally-supervised, transductive setup. Given a set of words, we label a small number of words with their correct label according to the category at hand (+1 for words that belong to the category, -1 for words that do not belong to it). Our model is based on an undirected weighted graph, in which vertices correspond to words, and edges correspond to relations between words. Our goal is to classify the unlabeled words (vertices) in the graph through a label propagation process. We now turn to describe our model in detail.

Graph Construction. We construct our graph such that an edge is added between two words (vertices) if both words participate in a symmetric pattern. The edge generation process is performed as follows. We first apply our symmetric pattern extraction procedure (Algorithm 1), and denote the set of selected symmetric patterns with P. We then traverse a large corpus[4] and extract all word pairs that participate in any pattern $p \in P$. We denote the number of occurrences of a word pair (w_1, w_2) in such patterns with f_{w_1,w_2}. Finally, we select all word pairs (w_1, w_2) for which $min(f_{w_1,w_2}, f_{w_2,w_1}) > \alpha$. Each such

[2] We set the maximal flexible pattern length to be 5.

[3] `https://books.google.com/ngrams`

[4] We use google books 5-grams (Michel et al., 2011).

Algorithm 1 Symmetric pattern extraction

```
procedure EXTRACT_SYMMETRIC_PATTERNS($C, W$)
    ▷ $C$ is a large corpus, $W$ is a lexicon
    ▷ Traverse $C$ and extract all flexible patterns of length 3-5 that appear in $C$ and contain exactly two content words
    $P \leftarrow$ extract_flexible_patterns($C, W$)
    for $p \in P$ do
        if $p$ appears in $<10^{-6}$ of the sentences in $C$ then
            Discard $p$ and continue
        end if
        $G_p \leftarrow$ a directed graph s.t. $V(G_p) \leftarrow W, E(G_p) \leftarrow \{(w_1, w_2) \in W^2 : w_1, w_2$ participate in at least one instance of $p\}$
        ▷ An undirected graph based on the bidirectional edges of the $G_p$
        $symG_p \leftarrow$ an undirected graph: $\{(w_1), (w_1, w_2) : (w_1, w_2) \in E(G_p) \wedge (w_2, w_1) \in E(G_p)\}$
        ▷ Two measures of symmetry
        $M_1 \leftarrow \frac{|V(symG_p)|}{|V(G_p)|}, M_2 \leftarrow \frac{|E(symG_p)|}{|E(G_p)|}$
        ▷ Symmetric pattern candidates are those with high $M_1$ and $M_2$ values
        if $\min(M_1, M_2) < 0.05$ then
            Discard $p$
        end if
    end for
    for $p \in P$ do
        ▷ E.g., "CW and CW" is contained in "both CW and CW"
        if $\exists p' \in P$ s.t. $p'$ is contained in $p$ then
            Discard $p$
        end if
    end for
    return The top 20 members of $P$ with the highest $M_1$ value
end procedure
```

pair is connected with an edge e_{w_1,w_2} in the graph, where the edge weight (denoted with w_{w_1,w_2}) is the geometric mean between f_{w_1,w_2} and f_{w_2,w_1}.

Label Propagation. Given a small number of annotated words (vertices), our goal is to propagate the information these words convey to other words in the graph. To do so, we apply an iterative variant of the k-Nearest Neighbors algorithm (I-k-NN). This iterative variant is required due to graph sparsity; when starting with a small set of labeled vertices, most unlabeled vertices do not have any labeled neighbor, and thus running the standard k-NN algorithm would result in classifying a very small number of vertices. Our approach is to run iterations of the k-NN algorithm, and thus propagate information to additional vertices at each iteration. At each k-NN step, the algorithm selects words that have at least one labeled neighbor. From this set, only the words that have the highest ratio of neighbors with the same label are selected, and are assigned with this label.

Consider a simple example. Say we have three candidate vertices a, b and c, where a has one neighbor with label +1 ($ratio(a) = 1/1 = 1.0$), b has two neighbors with label -1 ($ratio(b) = 2/2 = 1.0$) and c has three neighbors with label +1 and one neighbor with label -1 ($ratio(c) = max(3,1)/4 = 3/4$). Then, a and b are selected and are assigned with $+1$ and -1, respectively.

Seed Expansion. In minimally supervised setups like ours, the model is initialized with a small set of labeled seed examples. A natural approach in such settings is to apply a seed expansion step, in order to obtain a larger set of labeled seeds. Our method uses the same graph construction procedure described above, but uses a larger edge generation threshold $\beta >> \alpha$.[5] We then apply an iterative procedure that labels a vertex v with a label l if either (a) v is directly connected to γ of the vertices labeled with l or (b) v is connected to δ_l of the neighbors of vertices labeled with l.[6] This procedure is run iteratively until no more vertices meet any of the criteria (a) or (b).

[5] Using a larger threshold results in a sparser graph. Nevertheless, each edge in this graph is more likely to represent a real semantic relation.

[6] γ and δ_l are hyperparameters tuned on our development set (see Section 5.2).

5 Experimental Setup

5.1 Baselines

We compare our model to two types of baselines. The first (Classification Features Baselines) utilizes the I-k-NN algorithm, along with a different set of classification features. The second (Label Propagation Baselines) utilizes the same classification features as we do, but replaces I-k-NN with a more sophisticated label propagation algorithm.

5.1.1 Classification Features Baselines

In this set of baselines, we use different methods for building our graph. Concretely, instead of adding edges for pairs of words that appear in the same symmetric pattern, we use word similarity measures based on different feature sets as described below. The process of building the graph using the baseline word similarity measures is described in Section 5.2.

SENNA. Deep neural networks have gained recognition as leading feature extraction methods for word representation (Collobert and Weston, 2008; Socher et al., 2013). We use SENNA,[7] a deep network based word embedding method, which has been used to produce state-of-the-art results in several NLP tasks, including POS tagging, chunking, NER, parsing and SRL (Collobert et al., 2011). We use the cosine similarity between two word embeddings as a word similarity measure.

Brown. This baseline is derived from the clustering induced by the Brown algorithm (Brown et al., 1992).[8] This clustering, in which words share a cluster if they tend to appear in the same lexical context, has shown useful for several NLP tasks, including POS tagging (Clark, 2000), NER (Miller et al., 2004) and dependency parsing (Koo et al., 2008). We use it in order to control for the possibility that a simple contextual preference similarity correlates with similarity in semantic categorization better than symmetric pattern features.

The Brown algorithm builds a binary tree, where words are located at leaf nodes. We use the graph distance between two words u, v (i.e., the shortest path length between u, v in the tree) as a word similarity measure for building our graph.

5.1.2 Label Propagation Baselines

In this type of baselines, we replace I-k-NN with a different label propagation algorithm, while still using the symmetric pattern features for word similarity computations.

N-Cut. This baseline applies the normalized graph cut algorithm (Yu and Shi, 2003)[9] for label propagation. Given a graph $G = (V, E)$ and two sets of vertices $A, B \subseteq V$, this algorithm defines $links(A, B)$ to be the sum of edge weights between A and B. The objective of the algorithm is to find the clusters $A, V \setminus A$ that minimize $\frac{links(A,V \setminus A)}{links(A,V)}$. The algorithm of (Yu and Shi, 2003) is particularly efficient for this problem as it avoids eigenvector computations which may become computationally prohibitive for large graphs (for more details, see their paper). In order to encode information about our labeled seed words, we hard-code a large negative value (-100000) to the weights of edges between seed words with different labels (positive and negative).

MAD. The Modified Adsorption (MAD) algorithm (Talukdar and Crammer, 2009)[10] is an extension of the Adsorption algorithm (Baluja et al., 2008). MAD is a stochastic graph-based label propagation algorithm which has shown to have a number of attractive theoretical properties and demonstrated good experimental results.

[7] The word embeddings were downloaded from `http://ml.nec-labs.com/senna/`

[8] We use the clusters induced by (Koo et al., 2008), who applied the Brown algorithm implementation of (Liang, 2005) to the BLLIP corpus (Charniak et al., 2000). `http://www.people.csail.mit.edu/maestro/papers/bllip-clusters.gz`

[9] `http://www.cis.upenn.edu/~jshi/software/Ncut_9.zip`

[10] `http://github.com/parthatalukdar/junto`

5.2 Experiments

Graph Construction. We experiment with the CSLB dataset (Devereux et al., 2013), consisting of 638 nouns, annotated with their semantic categories by thirty human subjects. We first omit all nouns that are annotated as having more than one sense, and use the remaining 603 nouns to build our graph. From these nouns, 146 nouns are annotated as animate, 115 as edible, 50 as wearable and 35 as tools.[11] We then discard nouns that have less than two neighbors, which results in a final set of 450 nouns (vertices).

The graphs used in the classification features baselines are different than those used by the models that use our symmetric pattern classification features, since the features define the graph structure (Section 4). In order to provide a meaningful comparison, we build graphs with the same number of vertices for each of these baselines. We do so by selecting the n edges with the highest weight, together with the set of vertices connected by these edges, such that the resulting graph has 450 vertices. Working with these sets of vertices is the optimal setting for these baselines, as the resulting graphs are the ones with the highest possible edge weights for graphs with 450 vertices.[12]

Parameter Tuning. In order to avoid adding additional labeled examples for the sake of parameter tuning, we set the hyperparameter values to the ones for which each model performs best on an auxiliary semantic classification task. Concretely, we experiment with a fifth semantic category (*audibility*),[13] which is not part of our evaluation setting, for parameter tuning. Note that this results in our model having the same hyperparamter values for all four classification tasks.

In order to ensure that the models assign all participating words with labels, we set α=3, where α is the minimal number of times a word pair should appear in the same symmetric pattern in order to have an edge in our graph (See Section 4). In our seed expansion procedure, where we search for seeds whose label is predicted with high confidence, only word pairs that appear at least β=50 times in the same symmetric pattern are assigned an edge in the graph. We set the seed expansion procedure parameters to be $\gamma = 0.6, \delta_{+1} = 0.5, \delta_{-1} = 0.2$.

Evaluation. For each classification task, we run experiments with 4, 10, 20 and 40 labeled seed words. In each setting, half of the labeled seed words are assigned a positive label and the other half are assigned a negative label. For each semantic category and labeled seed set size, we repeat our experiment 1000 times, each of which with a different set of randomly selected labeled seed examples, and report the average results. We report both accuracy (number of correct labels divided by number of vertices in the graph) and F1 score, which is the harmonic mean of p (the average precision across labels) and r (average recall across labels).

These two measures represent complementary aspects of our results. On the one hand, accuracy is the most natural classification performance measure. On the other hand, the number of positive labels is substantially smaller than the number of negative labels,[14] and thus this measure can be manipulated: a dummy model that always assigns the negative label gets a high accuracy. The F1 score controls against such models by assigning them low scores.

6 Results

Our experiments are designed to explore two main questions: (a) the value of symmetric patterns as semantic classification features, compared to state-of-the-art word clustering and embedding methods; and (b) the required complexity of an algorithm that can propagate information about semantic similarity. Particularly, we test the value of our simple I-k-NN algorithm compared to more sophisticated alternatives.

A Minimally Supervised Setting. Our first set of experiments is in a minimally supervised setting where only two positive and two negative examples are available for each binary classification task. This

[11] Some words are classified as belonging to more than one category (e.g., "chicken" is both animate and edible).

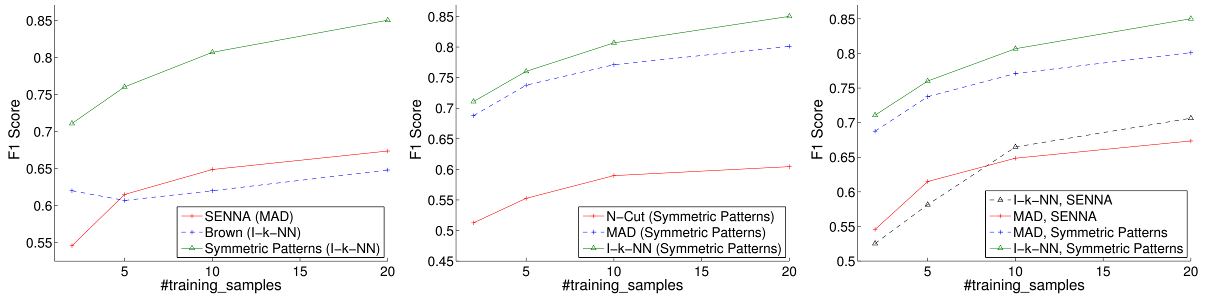
[12] The resulting graphs are actually denser than the symmetric patterns-based graph: 14K and 9K edges for the Brown and SENNA graphs, respectively, compared to $<$ 5K edges in the symmetric patterns graph.

[13] We used four labeled seed words in these experiments.

[14] Only 6-25% of the nouns have a positive label.

		Animacy			Edibility			is_worn			is_a_tool		
		SP	SENNA	Brown	SP	SENNA	Brown	SP	SENNA	Brown	SP	SENNA	Brown
Acc.	MAD	80.4%	77.7%	12.0%	75.0%	56.5%	14.8%	82.7%	66.8%	14.7%	73.3%	67.7%	12.2%
	N-Cut	71.4%	60.4%	51.2%	75.5%	59.4%	50.9%	83.3%	71.5%	51.4%	**82.7%**	77.1%	52.0%
	I-k-NN	**85.1%**	76.0%	55.5%	**82.2%**	56.8%	68.0%	**94.1%**	70.9%	66.7%	82.0%	75.7%	65.0%
F1	MAD	0.77	0.76	0.18	0.69	0.55	0.24	0.71	0.56	0.22	0.58	0.47	0.17
	N-Cut	0.49	0.45	0.46	0.51	0.44	0.45	0.61	0.56	0.41	0.56	0.50	0.38
	I-k-NN	**0.78**	0.70	0.48	**0.71**	0.53	0.62	**0.86**	0.59	0.55	**0.64**	0.52	0.51

Table 2: Accuracy and F1 score comparison between our model and the baselines. The columns correspond to the type of classification features used by the model: SP – symmetric patterns, SENNA – word embeddings extracted using deep networks (Collobert et al., 2011), Brown – Brown word clustering (Brown et al., 1992). The rows correspond to the algorithms applied by the model: N-Cut – the normalized graph cut algorithm (Yu and Shi, 2003), MAD – the modified adsorption algorithm (Talukdar and Crammer, 2009), I-k-NN – our iterative k-NN algorithm. Our model (I-k-NN + SP) is superior in all cases, except for the accuracy of the "is_a_tool" semantic category, where it is second only to N-Cut+SP.



(a) Classification Features Comparison (b) Algorithm Comparison (c) Top four Best Models

Figure 1: (a) Comparison of the different classification features. The figure shows the F1 scores of the best model that uses each of the feature sets (the label propagation algorithm used in each model appears in parentheses). (b) Comparison of the different label propagation algorithms. The figure shows the F1 scores of the best model that uses each of the algorithms (the classification feature sets used in each model appears in parentheses. It is always symmetric patterns). (c) The four best overall models (algorithm + classification feature set). The figures show that the symmetric pattern feature set is superior to the other feature sets, and that I-k-NN is superior or comparable to the other label propagation algorithms.

setup enables us to explore the performance of our model when the amounts of labeled training data is taken to the possible minimum.

Table 2 presents our results. With respect to objective (a), the table clearly demonstrates that symmetric patterns lead to much better results compared to the alternatives. Particularly, for all four semantic categories, and across both evaluation measures, it is a model that utilizes symmetric pattern classification features that achieves the best results. The average difference between the best model that uses symmetric patterns and the best model that does not is 12.5% accuracy and 0.13 F1 points. The dominance of symmetric pattern classification features is further demonstrated by the fact that a model that uses these features always performs better than a model that uses the same algorithm but different features.

With respect to objective (b) the table shows that I-k-NN provides a large improvement in seven out of eight (*category* × *evaluation measure*) settings. The average difference between the best model that utilizes I-k-NN and the best model that applies a different algorithm is 5.4% accuracy and 0.06 F1 points.

Analysis of Labeled Seed Set Size. In order to get a wider perspective on our model, we repeated our experiments with various sizes of the labeled seed set: 5,10 and 20 positive and negative labeled examples per semantic category. For brevity, only the F1 score results of the edibility category are presented. The trends observed on the other semantic categories (as well as when using the accuracy measure) are very similar.

Figure 1a compares the different classification features. For each feature f, results of the best performing model that uses f are shown. The results reveal that symmetric patterns clearly outperform the other features. The average differences between the best symmetric patterns-based model and the best models that use the other features are 0.15 (SENNA) and 0.16 (Brown) F1 points.

Figure 1b compares the different label propagation algorithms. For each algorithm a, results for the best performing model that uses a are presented. The results reveal that the I-k-NN algorithm outperforms both algorithms by 0.03 (MAD) and 0.21 (N-Cut) F1 points. The results also show that for all algorithms, the best performing model uses symmetric patterns classification features, which further demonstrates the dominance of these features.

Finally, Figure 1c presents the four top performing models (algorithm + classification feature). In accordance with the other findings presented in this section, the top two models, which outperform the other models by a large margin, apply symmetric pattern classification features.

Seed Expansion Effect. Our model uses a seed expansion procedure in order to expand a small set of labeled seed words to a larger set (see Section 4). In order to assess the quality of this procedure we compute, for each semantic category, the average size of the expanded set and the accuracy of the new seeds (i.e., the proportion of new seeds that are labeled correctly). Results show that the initial set is increased from four seeds (two positive + two negative) to 48-52, and that the accuracy of the new seeds is as high as 88-99%. Our experiments also show that this procedure provides a substantial performance boost to our I-k-NN algorithm, which obtains a 7.2% accuracy and 0.05 F1 points improvement (averaged over the four semantic categories) when applied with the expanded set of labeled seed words compared to the original set of size four.

7 Related Work

Classification into Semantic Categories. Several works tackled the task of semantic classification, mostly focusing on animacy, concreteness and countability. The vast majority of these works are either supervised (Hatzivassiloglou and McKeown, 1997; Baldwin and Bond, 2003; Peng and Araki, 2005; Øvrelid, 2005; Nagata et al., 2006; Xing et al., 2010; Kwong, 2011; Bowman and Chopra, 2012) or make use of external, language-specific resources such as WordNet (Orăsan and Evans, 2001; Orăsan and Evans, 2007; Moore et al., 2013). Our work, in contrast, is minimally supervised, requiring only a small set of labeled seed words.

Ji and Lin (2009) classified words into the gender and animacy categories, based on their occurrences in instances of hand-crafted patterns such as "X *who* Y" and "X *and his* Y". While their model uses patterns that are tailored to the animacy and gender categories, our model uses automatically induced patterns and is thus applicable to a range of semantic categories.

Finally, Turney et al. (2011) built a label propagation model that utilizes LSA (Landauer and Dumais, 1997) based classification features. They used their model to classify nouns into the concrete/abstract category using 40 labeled seed words . Unlike our model, which requires only a small set of labeled seeds, their algorithm is actually heavily supervised, requiring thousands of labeled examples for selecting the seed set of labeled words that are used for propagation. Our model, on the other hand, does not require any seed selection procedure, and utilizes a randomly selected set of labeled seed words.

Lexical Acquisition. Another line of work focused on the acquisition of semantic categories. In this setup, a model aims to find a core seed of words belonging to a given category, sacrificing recall for precision. Our model tackles a different task, namely the classification of words according to a given category where both recall and precision are to be optimized.

Lexical acquisition models are either supervised (Snow et al., 2006), unsupervised, making use of symmetric patterns (Davidov and Rappoport, 2006), or lightly supervised, requiring expert, language specific knowledge for compiling a set of hand-crafted patterns (Widdows and Dorow, 2002; Kozareva et al., 2008; Wang and Cohen, 2009). Other models require syntactic annotation derived from a supervised parser to extract coordination phrases (Riloff and Shepherd, 1997; Dorow et al., 2005). Our model automatically induces symmetric patterns, obtaining high quality results without relying on any type of language specific knowledge or annotation. Moreover, some of the works mentioned above (Riloff and Shepherd, 1997; Widdows and Dorow, 2002; Kozareva et al., 2008) also require manually selected label seeds to achieve good performance; in contrast, our work performs very well with a randomly selected set of labeled seed words.

8 Conclusion

We presented a minimally supervised model for noun classification into coarse grained semantic categories. Our model obtains 82%-94% accuracy on four semantic categories even when using only four labeled seed words per category. We showed that our modeling decisions – using symmetric patterns as classification features and a simple iterative k-NN algorithm for label propagation – lead to a substantial performance gain compared to state-of-the-art, more sophisticated, alternatives. Our results demonstrate the applicability of minimally supervised methods for semantic classification tasks. Future work will include modifying our model to support other, more fine-grained types of semantic categories, including adjectival categories (*properties*). We also plan to work on token-level word classification, and thus support multi-sense words, as well as demonstrate the power of unsupervised patterns acquisition for multilingual setups.

Acknowledgments

This research was funded (in part) by the Harry and Sylvia Hoffman leadership and responsibility program (for the first author), the Google Faculty research award (for the second author), the Intel Collaborative Research Institute for Computational Intelligence (ICRI-CI) and the Israel Ministry of Science and Technology Center of Knowledge in Machine Learning and Artificial Intelligence (Grant number 3-9243).

References

O. Abend and A. Rappoport. 2013. Universal conceptual cognitive annotation (UCCA). In *Proc. of ACL*.

T. Baldwin and F. Bond. 2003. A plethora of methods for learning English countability. In *Proc. of EMNLP*.

S. Baluja, R. Seth, D. Sivakumar, Y. Jing, J. Yagnik, S. Kumar, D. Ravichandran, and M. Aly. 2008. Video suggestion and discovery for youtube: taking random walks through the view graph. In *Proc. of WWW*, pages 895–904. ACM.

M. Baroni and A. Lenci. 2010. Distributional memory: A general framework for corpus-based semantics. *Computational Linguistics*, 36(4):673–721.

M. Berland and E. Charniak. 1999. Finding parts in very large corpora. In *Proc. of ACL*.

S. R. Bowman and H. Chopra. 2012. Automatic Animacy Classification. In *Proc. of NAACL-HLT Student Research Workshop*.

P. F. Brown, P. V. Desouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai. 1992. Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479.

A. M. Chan, J. M. Baker, E. Eskandar, D. Schomer, I. Ulbert, K. Marinkovic, S. S. Cash, and E. Halgren. 2011. First-pass selectivity for semantic categories in human anteroventral temporal lobe. *The Journal of Neuroscience*, 31(49):18119–18129.

E. Charniak, D. Blaheta, N. Ge, K. Hall, J. Hale, and M. Johnson. 2000. BLLIP 198789 WSJ Corpus Release 1, LDC No. LDC2000T43. Linguistic Data Consortium.

A. Clark. 2000. Inducing syntactic categories by context distribution clustering. In *Proc. of CoNLL*.

E. V. Clark. 2009. *First language acquisition*. Cambridge University Press.

R. Collobert and J. Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proc. of ICML*.

R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. 2011. Natural language processing (almost) from scratch. *JMLR*, 12:2493–2537.

D. Davidov and A. Rappoport. 2006. Efficient unsupervised discovery of word categories using symmetric patterns and high frequency words. In *Proc. of ACL-Coling*.

D. Davidov and A. Rappoport. 2008. Unsupervised discovery of generic relationships using pattern clusters and its evaluation by automatically generated SAT analogy questions. In *Proc. of ACL-HLT*.

B. J. Devereux, L. K. Tyler, J. Geertzen, and B. Randall. 2013. The centre for speech, language and the brain (CSLB) concept property norms. *Behavior research methods*, pages 1–9.

R. M. Dixon. 2005. *A semantic approach to English grammar*. Oxford University Press.

B. Dorow, D. Widdows, K. Ling, J. P. Eckmann, D. Sergi, and E. Moses. 2005. Using Curvature and Markov Clustering in Graphs for Lexical Acquisition and Word Sense Discrimination.

R. Evans and C. Orăsan. 2000. Improving anaphora resolution by identifying animate entities in texts. In *Proc. of DAARC*.

V. Hatzivassiloglou and K. R. McKeown. 1997. Predicting the semantic orientation of adjectives. In *Proc. of ACL*.

M. A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proc. of Coling – Volume 2*.

H. Ji and D. Lin. 2009. Gender and Animacy Knowledge Discovery from Web-Scale N-Grams for Unsupervised Person Mention Detection. In *Proc. of PACLIC*.

M. A. Just, V. L. Cherkassky, S. Aryal, and T. M. Mitchell. 2010. A neurosemantic theory of concrete noun representation based on the underlying brain codes. *PloS one*, 5(1):e8622.

T. Koo, X. Carreras, and M. Collins. 2008. Simple semi-supervised dependency parsing. In *Proc. of ACL-HLT*.

Z. Kozareva, E. Riloff, and E. Hovy. 2008. Semantic class learning from the web with hyponym pattern linkage graphs. In *Proc. of ACL-HLT*.

O. Y. Kwong. 2011. Measuring concept concreteness from the lexicographic perspective. In *Proc. of PACLIC*.

T. K. Landauer and S. T. Dumais. 1997. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, 104(2):211.

R. W. Langacker. 2008. *Cognitive grammar: A basic introduction*. Oxford University Press.

P. Liang. 2005. Semi-supervised learning for natural language. Master's thesis, Massachusetts Institute of Technology.

J. M. Mandler. 2004. *The foundations of mind: Origins of conceptual thought*. Oxford University Press New York.

A. Martin. 2007. The representation of object concepts in the brain. *Annual Review of Psychology*, 58:25–45.

J. B. Michel, Y. K. Shen, A. P. Aiden, A. Veres, M. K. Gray, J. P. Pickett, D. Hoiberg, D. Clancy, P. Norvig, J. Orwant, et al. 2011. Quantitative analysis of culture using millions of digitized books. *Science*, 331(6014):176–182.

S. Miller, J. Guinness, and A. Zamanian. 2004. Name tagging with word clusters and discriminative training. In *Proc. of NAACL*.

J. L. Moore, C. J. Burges, E. Renshaw, and W.-t. Yih. 2013. Animacy Detection with Voting Models. In *Proc. of EMNLP*.

R. Nagata, A. Kawai, K. Morihiro, and N. Isu. 2006. Reinforcing English countability prediction with one countability per discourse property. *Proc. of ACL-Coling*.

T. Naselaris, D. E. Stansbury, and J. L. Gallant. 2012. Cortical representation of animate and inanimate objects in complex natural scenes. *Journal of Physiology-Paris*, 106(5):239–249.

C. Orăsan and R. Evans. 2001. Learning to identify animate references. In *Proc. of the Workshop on Computational Natural Language*.

C. Orăsan and R. Evans. 2007. NP Animacy Identification for Anaphora Resolution. *JAIR*, 29:79–103.

L. Øvrelid. 2005. Animacy classification based on morphosyntactic corpus frequencies: some experiments with Norwegian nouns. In *Proc. of the Workshop on Exploring Syntactically Annotated Corpora*, pages 1–11.

J. Peng and K. Araki. 2005. Detecting the countability of english compound nouns using web-based models. In *Proc. of IJCNLP*.

E. Riloff and J. Shepherd. 1997. A corpus-based approach for building semantic lexicons. In *Proc. of EMNLP*.

R. Schwartz, O. Tsur, A. Rappoport, and M. Koppel. 2013. Authorship Attribution of Micro-Messages. In *Proc. of EMNLP*.

R. Snow, D. Jurafsky, and A. Y. Ng. 2006. Semantic taxonomy induction from heterogenous evidence. In *Proc. of ACL-Coling*.

R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proc. of EMNLP*.

G. Sudre, D. Pomerleau, M. Palatucci, L. Wehbe, A. Fyshe, R. Salmelin, and T. Mitchell. 2012. Tracking neural coding of perceptual and semantic features of concrete nouns. *NeuroImage*, 62(1):451–463.

P. P. Talukdar and K. Crammer. 2009. New regularized algorithms for transductive learning. In *ECML-PKDD*, pages 442–457. Springer.

O. Tsur, D. Davidov, and A. Rappoport. 2010. ICWSM – a great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews. In *Proc. of ICWSM*.

P. D. Turney and P. Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37(1):141–188.

P. Turney, Y. Neuman, D. Assaf, and Y. Cohen. 2011. Literal and metaphorical sense identification through concrete and abstract context. In *Proc. of EMNLP*.

P. D. Turney. 2008. The latent relation mapping engine: Algorithm and experiments. *Journal of Artificial Intelligence Research*, 33:615–655.

R. C. Wang and W. W. Cohen. 2009. Automatic set instance extraction using the web. In *Proc. of ACL-IJCNLP*.

D. Widdows and B. Dorow. 2002. A graph model for unsupervised lexical acquisition. In *Proc. of Coling*.

X. Xing, Y. Zhang, and M. Han. 2010. Query difficulty prediction for contextual image retrieval. In *Advances in Information Retrieval*, pages 581–585. Springer.

S. X. Yu and J. Shi. 2003. Multiclass spectral clustering. In *Proc. of ICCV*.

Published in CoNLL 2015

Chapter 3

Symmetric Pattern Based Word Embeddings for Improved Word Similarity Prediction

Symmetric Pattern Based Word Embeddings for Improved Word Similarity Prediction

Roy Schwartz,[1] **Roi Reichart,**[2] **Ari Rappoport**[1]
[1]Institute of Computer Science, The Hebrew University
[2]Technion, IIT
{roys02|arir}@cs.huji.ac.il roiri@ie.technion.ac.il

Abstract

We present a novel word level vector representation based on *symmetric patterns (SPs)*. For this aim we automatically acquire SPs (e.g., "X *and* Y") from a large corpus of plain text, and generate vectors where each coordinate represents the co-occurrence in SPs of the represented word with another word of the vocabulary. Our representation has three advantages over existing alternatives: First, being based on symmetric word relationships, it is highly suitable for word similarity prediction. Particularly, on the SimLex999 word similarity dataset, our model achieves a Spearman's ρ score of 0.517, compared to 0.462 of the state-of-the-art word2vec model. Interestingly, our model performs exceptionally well on verbs, outperforming state-of-the-art baselines by 20.2–41.5%. Second, pattern features can be adapted to the needs of a target NLP application. For example, we show that we can easily control whether the embeddings derived from SPs deem antonym pairs (e.g. (*big,small*)) as similar or dissimilar, an important distinction for tasks such as word classification and sentiment analysis. Finally, we show that a simple combination of the word similarity scores generated by our method and by word2vec results in a superior predictive power over that of each individual model, scoring as high as 0.563 in Spearman's ρ on SimLex999. This emphasizes the differences between the signals captured by each of the models.

1 Introduction

In the last decade, vector space modeling *(VSM)* for word representation (a.k.a word embedding), has become a key tool in NLP. Most approaches to word representation follow the distributional hypothesis (Harris, 1954), which states that words that co-occur in similar contexts are likely to have similar meanings.

VSMs differ in the way they exploit word co-occurrence statistics. Earlier works (see (Turney et al., 2010)) encode this information directly in the features of the word vector representation. More Recently, Neural Networks have become prominent in word representation learning (Bengio et al., 2003; Collobert and Weston, 2008; Collobert et al., 2011; Mikolov et al., 2013a; Pennington et al., 2014, inter alia). Most of these models aim to learn word vectors that maximize a language model objective, thus capturing the tendencies of the represented words to co-occur in the training corpus. VSM approaches have resulted in highly useful word embeddings, obtaining high quality results on various semantic tasks (Baroni et al., 2014).

Interestingly, the impressive results of these models are achieved despite the shallow linguistic information most of them consider, which is limited to the tendency of words to co-occur together in a pre-specified context window. Particularly, very little information is encoded about the syntactic and semantic relations between the participating words, and, instead, a bag-of-words approach is taken.[1]

This bag-of-words approach, however, comes with a cost. As recently shown by Hill et al. (2014), despite the impressive results VSMs that take this approach obtain on modeling word *association*, they are much less successful in modeling word *similarity*. Indeed, when evaluating these VSMs with datasets such as wordsim353 (Finkelstein et al., 2001), where the word pair scores re-

[1] A few recent VSMs go beyond the bag-of-words assumption and consider deeper linguistic information in word representation. We address this line of work in Section 2.

flect association rather than similarity (and therefore the (*cup,coffee*) pair is scored higher than the (*car,train*) pair), the Spearman correlation between their scores and the human scores often crosses the 0.7 level. However, when evaluating with datasets such as SimLex999 (Hill et al., 2014), where the pair scores reflect similarity, the correlation of these models with human judgment is below 0.5 (Section 6).

In order to address the challenge in modeling word similarity, we propose an alternative, pattern-based, approach to word representation. In previous work patterns were used to represent a variety of semantic relations, including hyponymy (Hearst, 1992), meronymy (Berland and Charniak, 1999) and antonymy (Lin et al., 2003). Here, in order to capture similarity between words, we use *Symmetric patterns (SPs)*, such as "X *and* Y" and "X *as well as* Y", where each of the words in the pair can take either the *X* or the *Y* position. Symmetric patterns have shown useful for representing similarity between words in various NLP tasks including lexical acquisition (Widdows and Dorow, 2002), word clustering (Davidov and Rappoport, 2006) and classification of words to semantic categories (Schwartz et al., 2014). However, to the best of our knowledge, they have not been applied to vector space word representation.

Our representation is constructed in the following way (Section 3). For each word w, we construct a vector v of size V, where V is the size of the lexicon. Each element in v represents the co-occurrence in SPs of w with another word in the lexicon, which results in a sparse word representation. Unlike most previous works that applied SPs to NLP tasks, we do not use a hard coded set of patterns. Instead, we extract a set of SPs from plain text using an unsupervised algorithm (Davidov and Rappoport, 2006). This substantially reduces the human supervision our model requires and makes it applicable for practically every language for which a large corpus of text is available.

Our SP-based word representation is flexible. Particularly, by exploiting the semantics of the pattern based features, our representation can be adapted to fit the specific needs of target NLP applications. In Section 4 we exemplify this property through the ability of our model to control whether its word representations will deem antonyms similar or dissimilar. *Antonyms* are words that have opposite semantic meanings (e.g., (*small,big*)), yet, due to their tendency to co-occur in the same context, they are often assigned similar vectors by co-occurrence based representation models (Section 6). Controlling the model judgment of antonym pairs is highly useful for NLP tasks: in some tasks, like word classification, antonym pairs such as (*small,big*) belong to the same class (size adjectives), while in other tasks, like sentiment analysis, identifying the difference between them is crucial. As discussed in Section 4, we believe that this flexibility holds for various other pattern types and for other lexical semantic relations (e.g. hypernymy, the *is-a* relation, which holds in word pairs such as (*dog,animal*)).

We experiment (Section 6) with the SimLex999 dataset (Hill et al., 2014), consisting of 999 pairs of words annotated by human subjects for similarity. When comparing the correlation between the similarity scores derived from our learned representation and the human scores, our representation receives a Spearman correlation coefficient score (ρ) of 0.517, outperforming six strong baselines, including the state-of-the-art *word2vec* (Mikolov et al., 2013a) embeddings, by 5.5–16.7%. Our model performs particularly well on the verb portion of SimLex999 (222 verb pairs), achieving a Spearman score of 0.578 compared to scores of 0.163–0.376 of the baseline models, an astonishing improvement of 20.2–41.5%. Our analysis reveals that the antonym adjustment capability of our model is vital for its success.

We further demonstrate that the word pair scores produced by our model can be combined with those of word2vec to get an improved predictive power for word similarity. The combined scores result in a Spearman's ρ correlation of 0.563, a further 4.6% improvement compared to our model, and a total of 10.1–21.3% improvement over the baseline models. This suggests that the models provide complementary information about word semantics.

2 Related Work

Vector Space Models for Lexical Semantics. Research on vector spaces for word representation dates back to the early 1970's (Salton, 1971). In traditional methods, a vector for each word w is generated, with each coordinate representing the co-occurrence of w and another context item of interest – most often a word but possibly also a sentence, a document or other items. The feature representation generated by this basic construction is sometimes post-processed using techniques such as Positive Pointwise Mutual Information (PPMI) normalization and dimensionality reduction. For recent surveys, see (Turney et al., 2010; Clark, 2012; Erk, 2012).

Most VSM works share two important characteristics. First, they encode co-occurrence statistics from an input corpus directly into the word vector features. Second, they consider very little information on the syntactic and semantic relations between the represented word and its context items. Instead, a bag-of-words approach is taken.

Recently, there is a surge of work focusing on Neural Network (NN) algorithms for word representations learning (Bengio et al., 2003; Collobert and Weston, 2008; Mnih and Hinton, 2009; Collobert et al., 2011; Dhillon et al., 2011; Mikolov et al., 2013a; Mnih and Kavukcuoglu, 2013; Lebret and Collobert, 2014; Pennington et al., 2014). Like the more traditional models, these works also take the bag-of-words approach, encoding only shallow co-occurrence information between linguistic items. However, they encode this information into their objective, often a language model, rather than directly into the features.

Consider, for example, the successful word2vec model (Mikolov et al., 2013a). Its continuous-bag-of-words architecture is designed to predict a word given its past and future context. The resulted objective function is:

$$\max \sum_{t=1}^{T} \log p(w_t | w_{t-c}, \ldots, w_{t-1}, w_{t+1}, \ldots, w_{t+c})$$

where T is the number of words in the corpus, and c is a pre-determined window size. Another word2vec architecture, skip-gram, aims to predict the past and future context given a word. Its objective is:

$$\max \sum_{t=1}^{T} \sum_{-c \leq j \leq c, j \neq 0} \log p(w_{t+j} | w_t)$$

In both cases the objective function relates to the co-occurrence of words within a context window.

A small number of works went beyond the bag-of-words assumption, considering deeper relationships between linguistic items. The Strudel system (Baroni et al., 2010) represents a word using the clusters of lexico-syntactic patterns in which it occurs. Murphy et al. (2012) represented words through their co-occurrence with other words in syntactic dependency relations, and then used the Non-Negative Sparse Embedding (NNSE) method to reduce the dimension of the resulted representation. Levy and Goldberg (2014) extended the skip-gram word2vec model with negative sampling (Mikolov et al., 2013b) by basing the word co-occurrence window on the dependency parse tree of the sentence. Bollegala et al. (2015) replaced bag-of-words contexts with various patterns (lexical, POS and dependency).

We introduce a symmetric pattern based approach to word representation which is particularly suitable for capturing word similarity. In experiments we show the superiority of our model over six models of the above three families: (a) bag-of-words models that encode co-occurrence statistics directly in features; (b) NN models that implement the bag-of-words approach in their objective; and (c) models that go beyond the bag-of-words assumption.

Similarity vs. Association Most recent VSM research does not distinguish between association and similarity in a principled way, although notable exceptions exist. Turney (2012) constructed two VSMs with the explicit goal of capturing either similarity or association. A classifier that uses the output of these models was able to predict whether two concepts are associated, similar or both. Agirre et al. (2009) partitioned the wordsim353 dataset into two subsets, one focused on similarity and the other on association. They demonstrated the importance of the association/similarity distinction by showing that some VSMs perform relatively well on one subset while others perform comparatively better on the other.

Recently, Hill et al. (2014) presented the SimLex999 dataset consisting of 999 word pairs judged by humans for similarity only. The participating words belong to a variety of POS tags and concreteness levels, arguably providing a more realistic sample of the English lexicon. Using their dataset the authors show the tendency of VSMs that take the bag-of-words approach to capture association much better than similarity. This observation motivates our work.

Symmetric Patterns. Patterns (*symmetric* or not) were found useful in a variety of NLP tasks, including identification of word relations such as hyponymy (Hearst, 1992), meronymy (Berland and Charniak, 1999) and antonymy (Lin et al., 2003). Patterns have also been applied to tackle sentence level tasks such as identification of sarcasm (Tsur et al., 2010), sentiment analysis (Davidov et al., 2010) and authorship attribution (Schwartz et al., 2013).

Symmetric patterns (SPs) were employed in various NLP tasks to capture different aspects of word similarity. Widdows and Dorow (2002) used SPs for the task of lexical acquisition. Dorow et al. (2005) and Davidov and Rappoport (2006) used them to perform unsupervised clustering of words. Kozareva et al. (2008) used SPs to classify proper names (e.g., fish names, singer names). Feng et al. (2013) used SPs to build a connotation lexicon, and Schwartz et al. (2014) used SPs to perform minimally supervised classification of words into semantic categories.

While some of these works used a hand crafted set of SPs (Widdows and Dorow, 2002; Dorow et al., 2005; Kozareva et al., 2008; Feng et al., 2013), Davidov and Rappoport (2006) introduced a fully unsupervised algorithm for the extraction of SPs. Here we apply their algorithm in order to reduce the required human supervision and demonstrate the language independence of our approach.

Antonyms. A useful property of our model is its ability to control the representation of antonym pairs. Outside the VSM literature several works identified antonyms using word co-occurrence statistics, manually and automatically induced patterns, the WordNet lexicon and thesauri (Lin et al., 2003; Turney, 2008; Wang et al., 2010; Mohammad et al., 2013; Schulte im Walde and Koper, 2013; Roth and Schulte im Walde, 2014). Recently, Yih et al. (2012), Chang et al. (2013) and Ono et al. (2015) proposed word representation methods that assign dissimilar vectors to antonyms. Unlike our unsupervised model, which uses plain text only, these works used the WordNet lexicon and a thesaurus.

3 Model

In this section we describe our approach for generating pattern-based word embeddings. We start by describing symmetric patterns (SPs), continue to show how SPs can be acquired automatically from text, and, finally, explain how these SPs are used for word embedding construction.

3.1 Symmetric Patterns

Lexico-syntactic patterns are sequences of words and wildcards (Hearst, 1992). Examples of patterns include "X *such as* Y", "X *or* Y" and "X *is a* Y". When patterns are instantiated in text, wildcards are replaced by words. For example, the pattern "**X** *is a* **Y**", with the **X** and **Y** wildcards, can be instantiated in phrases like "**Guffy** *is a* **dog**".

Candidate	Examples of Instances
"X *of* Y"	"point *of* view", "years *of* age"
"X *the* Y"	"around *the* world", "over *the* past"
"X *to* Y"	"nothing *to* do", "like *to* see"
"X *and* Y"	"men *and* women", "oil *and* gas"
"X *in* Y"	"keep *in* mind", "put *in* place"
"X *of the* Y"	"rest *of the* world", "end *of the* war"

Table 1:
The six most frequent pattern candidates that contain exactly two wildcards and 1-3 words in our corpus.

Symmetric patterns are a special type of patterns that contain exactly two wildcards and that tend to be instantiated by wildcard pairs such that each member of the pair can take the **X** or the **Y** position. For example, the symmetry of the pattern "**X** *or* **Y**" is exemplified by the semantically plausible expressions "cats *or* dogs" and "dogs *or* cats".

Previous works have shown that words that co-occur in SPs are semantically similar (Section 2). In this work we use symmetric patterns to represent words. Our hypothesis is that such representation would reflect word similarity (i.e., that similar vectors would represent similar words). Our experiments show that this is indeed the case.

Symmetric Patterns Extraction. Most works that used SPs manually constructed a set of such patterns. The most prominent patterns in these works are "X *and* Y" and "X *or* Y" (Widdows and Dorow, 2002; Feng et al., 2013). In this work we follow (Davidov and Rappoport, 2006) and apply an unsupervised algorithm for the automatic extraction of SPs from plain text.

This algorithm starts by defining an SP template to be a sequence of 3-5 tokens, consisting of exactly two wildcards, and 1-3 words. It then traverses a corpus, looking for frequent pattern candidates that match this template. Table 1 shows the six most frequent pattern candidates, along with common instances of these patterns.

The algorithm continues by traversing the pattern candidates and selecting a pattern p if a large portion of the pairs of words w_i, w_j that co-occur in p co-occur both in the ($X = w_i$,$Y = w_j$) form and in the ($X = w_j$,$Y = w_i$) form. Consider, for example, the pattern candidate "X *and* Y", and the pair of words "cat","dog". Both pattern instances "cat *and* dog" and "dog *and* cat" are likely to be seen in a large corpus. If this property holds for a large portion[2] of the pairs of words that co-occur in this pattern, it is selected as symmetric. On the other hand, the pattern candidate "X *of* Y" is in fact asymmetric: pairs of words such as "point", "view" tend to come only in the (X = "point",Y = "view") form and not the other way around. The reader is referred to (Davidov and Rappoport, 2006) for a more formal description of this algorithm. The resulting pattern set we use in this paper is "X *and* Y", "X *or* Y", "X *and the* Y", "*from* X *to* Y", "X *or the* Y", "X *as well as* Y", "X *or a* Y","X *rather than* Y", "X *nor* Y", "X *and one* Y", "*either* X *or* Y".

3.2 SP-based Word Embeddings

In order to generate word embeddings, our model requires a large corpus C, and a set of SPs P. The model first computes a symmetric matrix M of size $V \times V$ (where V is the size of the lexicon). In this matrix, $M_{i,j}$ is the co-occurrence count of both w_i,w_j and w_j,w_i in all patterns $p \in P$. For example, if w_i,w_j co-occur 1 time in p_1 and 3 times in p_5, while w_j,w_i co-occur 7 times in p_9, then $M_{i,j} = M_{j,i} = 1 + 3 + 7 = 11$. We then compute the Positive Pointwise Mutual Information (PPMI) of M, denoted by M^*.[3] The vector representation of the word w_i (denoted by v_i) is the i^{th} row in M^*.

Smoothing. In order to decrease the sparsity of our representation, we apply a simple smoothing technique. For each word w_i, W_i^n denotes the top n vectors with the smallest cosine-distance from v_i. We define the word embedding of w_i to be

$$v'_i = v_i + \alpha \cdot \sum_{v \in W_i^n} v$$

where α is a smoothing factor.[4] This process reduces the sparsity of our vector representation. For example, when $n = 0$ (i.e., no smoothing), the average number of non-zero values per vector is only 0.3K (where the vector size is ~250K). When $n = 250$, this number reaches ~14K.

4 Antonym Representation

In this section we show how our model allows us to adjust the representation of pairs of antonyms to the needs of a subsequent NLP task. This property will later be demonstrated to have a substantial impact on performance.

Antonyms are pairs of words with an opposite meaning (e.g., (*tall,short*)). As the members of an antonym pair tend to occur in the same context, their word embeddings are often similar. For example, in the skip-gram model (Mikolov et al., 2013a), the score of the (*accept,reject*) pair is 0.73, and the score of (*long,short*) is 0.71. Our SP-based word embeddings also exhibit a similar behavior.

The question of whether antonyms are similar or not is not a trivial one. On the one hand, some NLP tasks might benefit from representing antonyms as similar. For example, in word classification tasks, words such as "big" and "small" potentially belong to the same class (size adjectives), and thus representing them as similar is desired. On the other hand, antonyms are very dissimilar by definition. This distinction is crucial in tasks such as search, where a query such as "tall buildings" might be poorly processed if the representations of "tall" and "short" are similar.

In light of this, we construct our word embeddings to be *controllable* of antonyms. That is, our model contains an antonym parameter that can be turned on in order to generate word embeddings that represent antonyms as dissimilar, and turned off to represent them as similar.

To implement this mechanism, we follow (Lin et al., 2003), who showed that two patterns are particularly indicative of antonymy – "*from* X *to* Y" and "*either* X *or* Y" (e.g., "*from* **bottom** *to* **top**", "*either* **high** *or* **low**"). As it turns out, these two patterns are also symmetric, and are discovered by our automatic algorithm. Henceforth, we refer to these two patterns as *antonym patterns*.

Based on this observation, we present a variant of our model, which is designed to assign dissimilar vector representations to antonyms. We define two new matrices: M^{SP} and M^{AP}, which are computed similarly to M^* (see Section 3.2), only with different SP sets. M^{SP} is computed using the original set of SPs, excluding the two antonym patterns, while M^{AP} is computed using the two antonym patterns only.

Then, we define an antonym-sensitive, co-

[2] We use 15% of the pairs of words as a threshold.

[3] PPMI was shown useful for various co-occurrence models (Baroni et al., 2014).

[4] We tune n and α using a development set (Section 5). Typical values for n and α are 250 and 7, respectively.

occurrence matrix M^{+AN} to be

$$M^{+AN} = M^{SP} - \beta \cdot M^{AP}$$

where β is a weighting parameter.[5] Similarly to M^*, the antonym-sensitive word representation of the i^{th} word is the i^{th} row in M^{+AN}.

Discussion. The case of antonyms presented in this paper is an example of one relation that a pattern based representation model can control. This property can be potentially extended to additional word relations, as long as they can be identified using patterns. Consider, for example, the hypernymy relation (is-a, as in the (*apple,fruit*) pair). This relation can be accurately identified using patterns such as "X *such as* Y" and "X *like* Y" (Hearst, 1992). Consequently, it is likely that a pattern-based model can be adapted to control its predictions with respect to this relation using a method similar to the one we use to control antonym representation. We consider this a strong motivation for a deeper investigation of pattern-based VSMs in future work.

We next turn to empirically evaluate the performance of our model in estimating word similarity.

5 Experimental Setup

5.1 Datasets

Evaluation Dataset. We experiment with the SimLex999 dataset (Hill et al., 2014),[6] consisting of 999 pairs of words. Each pair in this dataset was annotated by roughly 50 human subjects, who were asked to score the similarity between the pair members. SimLex999 has several appealing properties, including its size, part-of-speech diversity, and diversity in the level of concreteness of the participating words.

We follow a 10-fold cross-validation experimental protocol. In each fold, we randomly sample 25% of the SimLex999 word pairs ($\sim$250 pairs) and use them as a development set for parameter tuning. We use the remaining 75% of the pairs ($\sim$750 pairs) as a test set. We report the average of the results we got in the 10 folds.

Training Corpus. We use an 8G words corpus, constructed using the word2vec script.[7] Through this script we also apply a pre-processing step which employs the word2phrase tool (Mikolov et al., 2013c) to merge common word pairs and triples to expression tokens. Our corpus consists of four datasets: (a) The 2012 and 2013 crawled news articles from the ACL 2014 workshop on statistical machine translation (Bojar et al., 2014);[8] (b) The One Billion Word Benchmark of Chelba et al. (2013);[9] (c) The UMBC corpus (Han et al., 2013);[10] and (d) The September 2014 dump of the English Wikipedia.[11]

5.2 Baselines

We compare our model against six baselines: one that encodes bag-of-words co-occurrence statistics into its features (model 1 below), three NN models that encode the same type of information into their objective function (models 2-4), and two models that go beyond the bag-of-words assumption (models 5-6). Unless stated otherwise, all models are trained on our training corpus.

1. BOW. A simple model where each coordinate corresponds to the co-occurrence count of the represented word with another word in the training corpus. The resulted features are re-weighted according to PPMI. The model's window size parameter is tuned on the development set.[12]

2-3. word2vec. The state-of-the-art *word2vec* toolkit (Mikolov et al., 2013a)[13] offers two word embedding architectures: continuous-bag-of-words (**CBOW**) and **skip-gram**. We follow the recommendations of the word2vec script for setting the parameters of both models, and tune the window size on the development set.[14]

4. GloVe. GloVe (Pennington et al., 2014)[15] is a global log-bilinear regression model for word embedding generation, which trains only on the nonzero elements in a co-occurrence matrix. We use the parameters suggested by the authors, and tune the window size on the development set.[16]

[5] We tune β using a development set (Section 5). Typical values are 7 and 10.

[6] `www.cl.cam.ac.uk/~fh295/simlex.html`

[7] `code.google.com/p/word2vec/source/browse/trunk/demo-train-big-model-v1.sh`

[8] `http://www.statmt.org/wmt14/training-monolingual-news-crawl/`

[9] `http://www.statmt.org/lm-benchmark/1-billion-word-language-modeling-benchmark-r13output.tar.gz`

[10] `http://ebiquity.umbc.edu/redirect/to/resource/id/351/UMBC-webbase-corpus`

[11] `dumps.wikimedia.org/enwiki/latest/enwiki-latest-pages-articles.xml.bz2`

[12] The value 2 is almost constantly selected.

[13] `https://code.google.com/p/word2vec/`

[14] Window size 2 is generally selected for both models.

[15] `nlp.stanford.edu/projects/glove/`

[16] Window size 2 is generally selected.

5. NNSE. The NNSE model (Murphy et al., 2012). As no full implementation of this model is available online, we use the off-the-shelf embeddings available at the authors' website,[17] taking the *full document and dependency* model with 2500 dimensions. Embeddings were computed using a dataset about twice as big as our corpus.

6. Dep. The modified, dependency-based, skip-gram model (Levy and Goldberg, 2014). To generate dependency links, we use the Stanford POS Tagger (Toutanova et al., 2003)[18] and the MALT parser (Nivre et al., 2006).[19] We follow the parameters suggested by the authors.

5.3 Evaluation

For evaluation we follow the standard VSM literature: the score assigned to each pair of words by a model m is the cosine similarity between the vectors induced by m for the participating words. m's quality is evaluated by computing the Spearman correlation coefficient score (ρ) between the ranking derived from m's scores and the one derived from the human scores.

6 Results

Main Result. Table 2 presents our results. Our model outperforms the baselines by a margin of 5.5–16.7% in the Spearman's correlation coefficient (ρ). Note that the capability of our model to control antonym representation has a substantial impact, boosting its performance from $\rho = 0.434$ when the antonym parameter is turned off to $\rho = 0.517$ when it is turned on.

Model Combination. We turn to explore whether our pattern-based model and our best baseline, skip-gram, which implements a bag-of-words approach, can be combined to provide an improved predictive power.

For each pair of words in the test set, we take a linear combination of the cosine similarity score computed using our embeddings and the score computed using the skip-gram (SG) embeddings:

$$f^{+}(w_i, w_j) = \gamma \cdot f_{SP}(w_i, w_j) + (1-\gamma) \cdot f_{SG}(w_i, w_j)$$

In this equation $f_{<m>}(w_i, w_j)$ is the cosine similarity between the vector representations of words w_i and w_j according to model m, and γ is a weighting parameter tuned on the development set (a common value is 0.8).

Model	**Spearman's ρ**
GloVe	0.35
BOW	0.423
CBOW	0.43
Dep	0.436
NNSE	0.455
skip-gram	0.462
SP$^{(-)}$	0.434
SP$^{(+)}$	**0.517**
Joint (SP$^{(+)}$, skip-gram)	**0.563**
Average Human Score	0.651

Table 2:
Spearman's ρ scores of our SP-based model with the antonym parameter turned on (*SP*$^{(+)}$) or off (*SP*$^{(-)}$) and of the baselines described in Section 5.2. *Joint (SP*$^{(+)}$*, skip-gram)* is an interpolation of the scores produced by *skip-gram* and our *SP*$^{(+)}$ model. *Average Human Score* is the average correlation of a single annotator with the average score of all annotators, taken from (Hill et al., 2014).

As shown in Table 2, this combination forms the top performing model on SimLex999, achieving a Spearman's ρ score of 0.563. This score is 4.6% higher than the score of our model, and a 10.1–21.3% improvement compared to the baselines.

wordsim353 Experiments. The wordsim353 dataset (Finkelstein et al., 2001) is frequently used for evaluating word representations. In order to be compatible with previous work, we experiment with this dataset as well. As our word embeddings are designed to support word similarity rather than relatedness, we focus on the similarity subset of this dataset, according to the division presented in (Agirre et al., 2009).

As noted by (Hill et al., 2014), the word pair scores in both subsets of wordsim353 reflect word association. This is because the two subsets created by (Agirre et al., 2009) keep the original wordsim353 scores, produced by human evaluators that were instructed to score according to association rather than similarity. Consequently, we expect our model to perform worse on this dataset compared to a dataset, such as SimLex999, whose annotators were guided to score word pairs according to similarity.

Contrary to SimLex999, wordsim353 treats antonyms as similar. For example, the similarity score of the (*life,death*) and (*profit,loss*) pairs are 7.88 and 7.63 respectively, on a 0-10 scale. Consequently, we turn the antonym parameter off for this experiment.

Table 3 presents the results. As expected, our

[17] `http://www.cs.cmu.edu/~bmurphy/NNSE/`

[18] `nlp.stanford.edu/software/`

[19] `http://www.maltparser.org/index.html`

Model	Spearman's ρ
GloVe	0.677
Dep	0.712
BOW	0.729
CBOW	0.734
NNSE	0.78
skip-gram	**0.792**
SP$^{(-)}$	0.728
Average Human Score	0.756

Table 3:
Spearman's ρ scores for the similarity portion of wordsim353 (Agirre et al., 2009). SP$^{(-)}$ is our model with the antonym parameter turned off. Other abbreviations are as in Table 2.

Model	Adj.	Nouns	Verbs
GloVe	0.571	0.377	0.163
Dep	0.54	0.449	0.376
BOW	0.548	0.451	0.276
CBOW	0.579	0.48	0.252
NNSE	0.594	0.487	0.318
skip-gram	0.604	**0.501**	0.307
SP$^{(+)}$	**0.663**	0.497	**0.578**

Table 4:
A POS-based analysis of the various models. Numbers are the Spearman's ρ scores of each model on each of the respective portions of SimLex999.

model is not as successful on a dataset that doesn't reflect pure similarity. Yet, it still crosses the $\rho = 0.7$ score, a quite high performance level.

Part-of-Speech Analysis. We next perform a POS-based evaluation of the participating models, using the three portions of the SimLex999: 666 pairs of nouns, 222 pairs of verbs, and 111 pairs of adjectives. Table 4 indicates that our SP$^{(+)}$ model is exceptionally successful in predicting verb and adjective similarity. On verbs, SP$^{(+)}$ obtains a score of $\rho = 0.578$, a 20.2–41.5% improvement over the baselines. On adjectives, SP$^{(+)}$ performs even better ($\rho = 0.663$), an improvement of 5.9–12.3% over the baselines. On nouns, SP$^{(+)}$ is second only to *skip-gram*, though with very small margin (0.497 vs. 0.501), and is outperforming the other baselines by 1–12%. The lower performance of our model on nouns might partially explain its relatively low performance on wordsim353, which is composed exclusively of nouns.

Analysis of Antonyms. We now turn to a qualitative analysis, in order to understand the impact of our modeling decisions on the scores of antonym word pairs. Table 5 presents examples of antonym pairs taken from the SimLex999 dataset, along with their relative ranking among all pairs in the set, as judged by our model (SP$^{(+)}$ with $\beta = 10$ or SP$^{(-)}$ with $\beta = -1$) and by the best baseline representation (skip-gram). Each pair of words is assigned a score between 1 and 10 by each model, where a score of M means that the pair is ranked at the M'th decile. The examples in the table are taken from the first (lowest) decile according to SimLex999's human evaluators. The table shows that when the antonym parameter is off, our model generally recognizes antonyms as similar. In contrast, when the parameter is on, ranks of antonyms substantially decrease.

Pair of Words	SP		*skip-gram*
	+AN	-AN	
new - old	1	6	6
narrow - wide	1	7	8
necessary - unnecessary	2	2	9
bottom - top	3	8	10
absence - presence	4	7	9
receive - send	1	9	8
fail - succeed	1	8	6

Table 5:
Examples of antonym pairs and their decile in the similarity ranking of our *SP* model with the antonym parameter turned on (+AN, β=10) or off (-AN, β=-1), and of the skip-gram model, the best baseline. All examples are judged in the lowest decile (1) by SimLex999's annotators.

Antonymy as Word Analogy. One of the most notable features of the skip-gram model is that some geometric relations between its vectors translate to semantic relations between the represented words (Mikolov et al., 2013c), e.g.:

$$v_{woman} - v_{man} + v_{king} \approx v_{queen}$$

It is therefore possible that a similar method can be applied to capture antonymy – a useful property that our model was demonstrated to have.

To test this hypothesis, we generated a set of 200 analogy questions of the form ”X - Y + Z = ?” where X and Y are antonyms, and Z is a word with an unknown antonym.[20] Example questions include: "*stupid - smart + life* = ?" (*death*) and "*huge - tiny + arrive* = ?" (*leave*). We applied the standard word analogy evaluation (Mikolov et al., 2013c) on this dataset with the skip-gram embeddings, and found that results are quite poor: 3.5% accuracy (compared to an average 56% accuracy this model obtains on a standard word analogy dataset (Mikolov et al., 2013a)). Given these results, the question of whether skip-gram is capa-

[20] Two human annotators selected a list of potential antonym pairs from SimLex999 and wordsim353. We took the intersection of their selections (26 antonym pairs) and randomly generated 200 analogy questions, each containing two antonym pairs. The dataset can be found in `www.cs.huji.ac.il/~roys02/papers/sp_embeddings/antonymy_analogy_questions.zip`

ble of accounting for antonyms remains open.

7 Conclusions

We presented a symmetric pattern based model for word vector representation. On SimLex999, our model is superior to six strong baselines, including the state-of-the-art word2vec skip-gram model by as much as 5.5–16.7% in Spearman's ρ score. We have shown that this gain is largely attributed to the remarkably high performance of our model on verbs, where it outperforms all baselines by 20.2–41.5%. We further demonstrated the adaptability of our model to antonym judgment specifications, and its complementary nature with respect to word2vec.

In future work we intend to extend our pattern-based word representation framework beyond symmetric patterns. As discussed in Section 4, other types of patterns have the potential to further improve the expressive power of word vectors. A particularly interesting challenge is to enhance our pattern-based approach with bag-of-words information, thus enjoying the provable advantages of both frameworks.

Acknowledgments

We would like to thank Elad Eban for his helpful advice. This research was funded (in part) by the Intel Collaborative Research Institute for Computational Intelligence (ICRI-CI) and the Israel Ministry of Science and Technology Center of Knowledge in Machine Learning and Artificial Intelligence (Grant number 3-9243).

References

Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Pasca, and Aitor Soroa. 2009. A study on similarity and relatedness using distributional and wordnet-based approaches. In *Proc. of HLT-NAACL*.

Marco Baroni, Brian Murphy, Eduard Barbu, and Massimo Poesio. 2010. Strudel: A corpus-based semantic model based on properties and types. *Cognitive Science*.

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proc. of ACL*.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *JMLR*.

Matthew Berland and Eugene Charniak. 1999. Finding parts in very large corpora. In *Proc. of ACL*.

Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, and Lucia Specia, editors. 2014. *Proc. of the Ninth Workshop on Statistical Machine Translation*.

Danushka Bollegala, Takanori Maehara, Yuichi Yoshida, and Ken ichi Kawarabayashi. 2015. Learning word representations from relational graphs. In *Proc. of AAAI*.

Kai-wei Chang, Wen-tau Yih, and Christopher Meek. 2013. Multi-Relational Latent Semantic Analysis. In *Proc. of EMNLP*.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, and Phillipp Koehn. 2013. One billion word benchmark for measuring progress in statistical language modeling. *CoRR*.

Stephen Clark. 2012. Vector space models of lexical meaning. *Handbook of Contemporary Semanticssecond edition*, pages 1–42.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proc. of ICML*.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *JMLR*, 12:2493–2537.

Dmitry Davidov and Ari Rappoport. 2006. Efficient unsupervised discovery of word categories using symmetric patterns and high frequency words. In *Proc. of ACL-Coling*.

Dmitry Davidov, Oren Tsur, and Ari Rappoport. 2010. Enhanced sentiment learning using twitter hashtags and smileys. In *Proc. of Coling*.

Paramveer Dhillon, Dean P Foster, and Lyle H Ungar. 2011. Multi-view learning of word embeddings via cca. In *Proc. of NIPS*.

Beate Dorow, Dominic Widdows, Katarina Ling, Jean-Pierre Eckmann, Danilo Sergi, and Elisha Moses. 2005. Using Curvature and Markov Clustering in Graphs for Lexical Acquisition and Word Sense Discrimination.

Katrin Erk. 2012. Vector Space Models of Word Meaning and Phrase Meaning: A Survey. *Language and Linguistics Compass*, 6(10):635–653.

Song Feng, Jun Seok Kang, Polina Kuznetsova, and Yejin Choi. 2013. Connotation lexicon: A dash of sentiment beneath the surface meaning. In *Proc. of ACL*.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2001. Placing search in context: The concept revisited. In *Proc. of WWW*.

Lushan Han, Abhay L. Kashyap, Tim Finin, James Mayfield, and Jonathan Weese. 2013. Umbc_ebiquity-core: Semantic textual similarity systems. In *Proc. of *SEM*.

Zellig Harris. 1954. Distributional structure. *Word*.

Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proc. of Coling – Volume 2*.

Felix Hill, Roi Reichart, and Anna Korhonen. 2014. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *arXiv:1408.3456 [cs.CL]*.

Zornitsa Kozareva, Ellen Riloff, and Eduard Hovy. 2008. Semantic class learning from the web with hyponym pattern linkage graphs. In *Proc. of ACL-HLT*.

Rémi Lebret and Ronan Collobert. 2014. Word embeddings through hellinger pca. In *Proc. of EACL*.

Omer Levy and Yoav Goldberg. 2014. Dependency-based word embeddings. In *Proc. of ACL (Volume 2: Short Papers)*.

Dekang Lin, Shaojun Zhao, Lijuan Qin, and Ming Zhou. 2003. Identifying synonyms among distributionally similar words. In *Proc. of IJCAI*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Proc. of NIPS*.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013c. Linguistic regularities in continuous space word representations. In *Proc. of NAACL-HLT*.

Andriy Mnih and Geoffrey E Hinton. 2009. A scalable hierarchical distributed language model. In *Proc. of NIPS*.

Andriy Mnih and Koray Kavukcuoglu. 2013. Learning word embeddings efficiently with noise-contrastive estimation. In *Proc. of NIPS*.

Saif M. Mohammad, Bonnie J. Dorr, Graeme Hirst, and Peter D. Turney. 2013. Computing lexical contrast. *Computational Linguistics*, 39(3):555–590.

Brian Murphy, Partha Pratim Talukdar, and Tom Mitchell. 2012. Learning Effective and Interpretable Semantic Models using Non-Negative Sparse Embedding. In *Proc. of Coling*.

Joakim Nivre, Johan Hall, and Jens Nilsson. 2006. Maltparser: A data-driven parser-generator for dependency parsing. In *Proc. of LREC*.

Masataka Ono, Makoto Miwa, and Yutaka Sasaki. 2015. Word embedding-based antonym detection using thesauri and distributional information. In *Proc. of NAACL*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proc. of EMNLP*.

Michael Roth and Sabine Schulte im Walde. 2014. Combining Word Patterns and Discourse Markers for Paradigmatic Relation Classification. In *Proc. of ACL*.

Gerard Salton. 1971. *The SMART Retrieval System: Experiments in Automatic Document Processing*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Sabine Schulte im Walde and Maximilian Koper. 2013. Pattern-based distinction of paradigmatic relations for german nouns, verbs, adjectives. *Language Processing and Knowledge in the Web*, pages 184–198.

Roy Schwartz, Oren Tsur, Ari Rappoport, and Moshe Koppel. 2013. Authorship attribution of micro-messages. In *Proc. of EMNLP*.

Roy Schwartz, Roi Reichart, and Ari Rappoport. 2014. Minimally supervised classification to semantic categories using automatically acquired symmetric patterns. In *Proc. of Coling*.

Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proc. of NAACL*.

Oren Tsur, Dmitry Davidov, and Ari Rappoport. 2010. Icwsm–a great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews. In *Proc. of ICWSM*.

Peter D. Turney, Patrick Pantel, et al. 2010. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence research*.

Peter D. Turney. 2008. A uniform approach to analogies, synonyms, antonyms, and associations. In *Proc. of Coling*.

Peter D. Turney. 2012. Domain and function: A dual-space model of semantic relations and compositions. *Journal of Artificial Intelligence Research*, pages 533–585.

Wenbo Wang, Christopher Thomas, Amit Sheth, and Victor Chan. 2010. Pattern-based synonym and antonym extraction. In *Proc. of ACM*.

Dominic Widdows and Beate Dorow. 2002. A graph model for unsupervised lexical acquisition. In *Proc. of Coling*.

Wen-tau Yih, Geoffrey Zweig, and John C. Platt. 2012. Polarity inducing latent semantic analysis. In *Proc. of EMNLP-CoNLL*.

Published in NAACL 2016

Chapter 4

Symmetric Patterns and Coordinations: Fast and Enhanced Representations of Verbs and Adjectives

Symmetric Patterns *and* Coordinations: Fast *and* Enhanced Representations of Verbs *and* Adjectives

Roy Schwartz,[1] **Roi Reichart,**[2] **Ari Rappoport**[1]
[1]Institute of Computer Science, The Hebrew University
[2]Faculty of Industrial Engineering and Management, Technion, IIT
{roys02|arir}@cs.huji.ac.il roiri@ie.technion.ac.il

Abstract

State-of-the-art word embeddings, which are often trained on bag-of-words (*BOW*) contexts, provide a high quality representation of aspects of the semantics of nouns. However, their quality decreases substantially for the task of verb similarity prediction. In this paper we show that using symmetric pattern contexts (*SPs*, e.g., "X and Y") improves word2vec verb similarity performance by up to 15% and is also instrumental in adjective similarity prediction. The unsupervised *SP* contexts are even superior to a variety of dependency contexts extracted using a supervised dependency parser. Moreover, we observe that *SPs* and dependency coordination contexts (*Coor*) capture a similar type of information, and demonstrate that *Coor* contexts are superior to other dependency contexts including the set of all dependency contexts, although they are still inferior to *SPs*. Finally, there are substantially fewer *SP* contexts compared to alternative representations, leading to a massive reduction in training time. On an 8G words corpus and a 32 core machine, the *SP* model trains in 11 minutes, compared to 5 and 11 hours with *BOW* and all dependency contexts, respectively.

1 Introduction

In recent years, vector space models (VSMs) have become prominent in NLP. VSMs are often evaluated by measuring their ability to predict human judgments of lexical semantic relations between pairs of words, mostly association or similarity. While many datasets for these tasks are limited to pairs of nouns, the recent SimLex999 word similarity dataset (Hill et al., 2014) also consists of similarity scores for *verb* and *adjective* pairs. State-of-the-art VSMs such as word2vec skip-gram (*w2v-SG*, (Mikolov et al., 2013a)) and GloVe (Pennington et al., 2014) excel at noun-related tasks. However, their performance substantially decreases on *verb* similarity prediction in SimLex999, and their adjective representations have rarely been evaluated (Section 2).

In this paper we show that a key factor in the reduced performance of the *w2v-SG* model on verb representation is its reliance on bag-of-words (*BOW*) contexts: contexts of the represented words that consist of words in their physical proximity. We investigate a number of alternative contexts for this model, including various dependency contexts, and show that simple, automatically acquired symmetric patterns (*SPs*, e.g., "X or Y", (Hearst, 1992; Davidov and Rappoport, 2006)) are the most useful contexts for the representation of verbs and also adjectives. Moreover, the *SP*-based model is much more compact than the alternatives, making its training an order of magnitude faster.

In particular, we train several versions of the *w2v-SG* model, each with a different context type, and evaluate the resulting word embeddings on the task of predicting the similarity scores of the verb and adjective portions of SimLex999. Our results show that *SP* contexts (SG-*SP*) obtain the best results on both tasks: Spearman's ρ scores of 0.459 on verbs and 0.651 on adjectives. These results are 15.2% and 4.7% better than *BOW* contexts and 7.3% and 6.5% better than all dependency contexts (*DepAll*). Moreover, the number of *SP* contexts is substantially smaller than the alternatives, making it extremely fast to train: 11 minutes only on an 8G word corpus using a 32 CPU core machine, compared to 5 and 11 hours for *BOW* and *DepAll*, respectively.

Recently, Schwartz et al. (2015) presented a count-based VSM that utilizes *SP* contexts (SRR15). This model excels on verb similarity, outperforming VSMs that use other contexts (e.g., *BOW* and *DepAll*) by more than 20%. In this paper we show that apart from its *SP* contexts, the success of SRR15 is attributed in large to its explicit representation of antonyms (*live/die*); turning this feature off reduces its performance to be on par with SG-*SP*. As opposed to Schwartz et al. (2015), we keep our VSM fixed across experiments (*w2v-SG*), changing only the context type. This allows us to attribute our improved results to one factor: *SP* contexts.

We further observe that *SP* contexts are tightly connected to syntactic *coordination* contexts (*Coor*, Section 3). Following this observation, we compare the *w2v-SG* model with three dependency-based context types: (a) *Coor* contexts; (b) all dependency links (*DepAll*); and (c) all dependency links excluding *Coor* links ($Coor^C$).[1] Our results show that training with *Coor* contexts is superior to training with the other context types, leading to improved similarity prediction of 2.7-4.1% and 4.3-6.9% on verbs and adjectives respectively.

These results demonstrate the prominence of *Coor* contexts in verb and adjective representation: these contexts are even better than their combination with the rest of the dependency-based contexts (the *DepAll* contexts). Nonetheless, although *Coor* contexts are extracted using a supervised dependency parser, they are still inferior to *SP* contexts, extracted automatically from plain text (Section 3), by 4.6% and 2.2% for verb and adjective pairs.

2 Background

Word Embeddings for Verbs and Adjectives. A number of evaluation sets consisting of word pairs scored by humans for semantic relations (mostly association and similarity) are in use for VSM evaluation. These include: RG-65 (Rubenstein and Goodenough, 1965), MC-30 (Miller and Charles, 1991), WordSim353 (Finkelstein et al., 2001), MEN (Bruni et al., 2014) and SimLex999 (Hill et al., 2014).[2]

Nouns are dominant in almost all of these datasets. For example, RG-65, MC-30 and WordSim353 consist of noun pairs almost exclusively. A few datasets contain pairs of verbs (Yang and Powers, 2006; Baker et al., 2014). The MEN dataset, although dominated by nouns, also contains verbs and adjectives. Nonetheless, the human judgment scores in these datasets reflect *relatedness* between words. In contrast, the recent SimLex999 dataset (Hill et al., 2014) contains word *similarity* scores for nouns (666 pairs), verbs (222 pairs) and adjectives (111 pairs). We use this dataset to study the effect of context type on VSM performance in a verb and adjective similarity prediction task.

Context Type in Word Embeddings. Most VSMs (e.g., (Collobert et al., 2011; Mikolov et al., 2013b; Pennington et al., 2014)) define the context of a target word to be the words in its physical proximity (bag-of-words contexts). Dependency contexts, consisting of the words connected to the target word by dependency links (Grefenstette, 1994; Padó and Lapata, 2007; Levy and Goldberg, 2014), are another well researched alternative. These works did not recognize the importance of syntactic coordination contexts (*Coor*).

Patterns have also been suggested as VSM contexts, but mostly for representing *pairs* of words (Turney, 2006; Turney, 2008). While this approach has been successful for extracting various types of word relations, using patterns to represent *single* words is useful for downstream applications. Recently, Schwartz et al. (2015) explored the value of *symmetric* pattern contexts for word representation, an idea this paper develops further.

A recently published approach (Melamud et al., 2016) also explored the effect of the type of context on the performance of word embedding models. Nonetheless, while they also explored bag-of-words and dependency contexts, they did not experiment with *SPs* or coordination contexts, which we find to be most useful for predicting word similarity.

Limitations of Word Embeddings. Recently, a few papers examined the limitations of word embedding models in representing different types of se-

[1] $Coor \cup Coor^C = DepAll$, $Coor \cap Coor^C = \emptyset$

[2] For a comprehensive list see: `wordvectors.org/`

mantic information. Levy et al. (2015) showed that word embeddings do not capture semantic relations such as hyponymy and entailment. Rubinstein et al. (2015) showed that while state-of-the-art embeddings are successful at capturing taxonomic information (e.g., *cow* is an animal), they are much less successful in capturing attributive properties (*bananas* are yellow). In (Schwartz et al., 2015), we showed that word embeddings are unable to distinguish between pairs of words with opposite meanings (antonyms, e.g., good/bad). In this paper we study the difficulties of bag-of-words based word embeddings in representing verb similarity.

3 Symmetric Patterns (*SPs*)

Lexico-syntactic patterns are templates of text that contain both words and wildcards (Hearst, 1992), e.g., "X *and* Y" and "X *for a* Y". Pattern *instances* are sequences of words that match a given pattern, such that concrete words replace each of the wildcards. For example, "**John** *and* **Mary**" is an instance of the pattern "X *and* Y". Patterns have been shown useful for a range of tasks, including word relation extraction (Lin et al., 2003; Davidov et al., 2007), knowledge extraction (Etzioni et al., 2005), sentiment analysis (Davidov et al., 2010) and authorship attribution (Schwartz et al., 2013).

Symmetric patterns (*SPs*) are lexico-syntactic patterns that comply to two constraints: (a) Each pattern has exactly two wildcards (e.g., **X** *or* **Y**); and (b) When two words (*X*,*Y*) co-occur in an *SP*, they are also likely to co-occur in this pattern in opposite positions, given a large enough corpus (e.g., "**X** *or* **Y**" and "**Y** *or* **X**"). For example, the pattern "**X** *and* **Y**" is symmetric as for a large number of word pairs (e.g., (*eat,drink*)) both members are likely to occur in both of its wildcard positions (e.g., "eat *and* drink", "drink *and* eat").

SPs have shown useful for tasks such as word clustering (Widdows and Dorow, 2002; Davidov and Rappoport, 2006), semantic class learning (Kozareva et al., 2008) and word classification (Schwartz et al., 2014). In this paper we demonstrate the value of *SP*-based contexts in vector representations of verbs and adjectives. The rationale behind this context type is that two words that co-occur in an *SP* tend to take the same semantic role in the sentence, and are thus likely to be similar in meaning (e.g., "(John and Mary) sang").

SP **Extraction.** Many works that applied *SPs* in NLP tasks employed a hand-crafted list of patterns (Widdows and Dorow, 2002; Dorow et al., 2005; Feng et al., 2013). Following Schwartz et al. (2015) we employ the DR06 algorithm (Davidov and Rappoport, 2006), an unsupervised algorithm that extracts *SPs* from plain text. We apply this algorithm to our corpus (Section 4) and extract 11 *SPs*: "X *and* Y", "X *or* Y", "X *and the* Y", "X *or the* Y", "X *or a* Y", "X *nor* Y", "X *and one* Y", "*either* X *or* Y", "X *rather than* Y", "X *as well as* Y", "*from* X *to* Y". A description of the DR06 algorithm is beyond the scope of this paper; the interested reader is referred to (Davidov and Rappoport, 2006).

SP **Contexts.** We generate *SP* contexts by taking the co-occurrence counts of pairs of words in *SPs*. For example, in the *SP* token "*boys and girls*", the term *girls* is taken as an *SP* context of the word *boys*, and *boys* is taken as an *SP* context of *girls*.

We do not make a distinction between the different *SPs*. E.g., "*boys* **and** *girls*" and "*boys* **or** *girls*" are treated the same. However, we distinguish between left and right contexts. For example, we generate different contexts for the word *girls*, one for left-hand contexts ("**girls** *and boys*") and another for right-hand contexts ("*boys and* **girls**").

SPs **and Coordinations.** *SPs* and syntactic coordinations (*Coors*) are intimately related. For example, of the 11 *SPs* extracted in this paper by the DR06 algorithm (listed above), the first eight represent coordination structures. Moreover, these *SPs* account for more than 98% of the *SP* instances in our corpus. Indeed, due to the significant overlap between *SPs* and *Coors*, the former have been proposed as a simple model of the latter (Nakov and Hearst, 2005).[3]

Despite their tight connection, *SPs* sometimes fail to properly identify the components of *Coors*. For example, while *SPs* are instrumental in capturing shallow *Coors*, they fail in capturing coordination between phrases. Consider the sentence *John*

[3] Note though that the exact syntactic annotation of coordination is debatable both in the linguistic community (Tesnière, 1959; Hudson, 1980; Mel'čuk, 1988) and also in the NLP community (Nilsson et al., 2006; Schwartz et al., 2011; Schwartz et al., 2012).

walked and Mary ran: the *SP* "X *and* Y" captures the phrase *walked and Mary*, while the *Coor* links the heads of the connected phrases ("*walked*" and "*ran*"). *SPs*, on the other hand, can go beyond *Coors* and capture other types of symmetric structures like "*from* X *to* Y" and "X *rather than* Y".

Our experiments reveal that both *SPs* and *Coors* are highly useful contexts for verb and adjective representation, at least with respect to word similarity. Interestingly, *Coor* contexts, extracted using a supervised dependency parser, are less effective than *SP* contexts, which are extracted from plain text.

4 Experiments

Model. We keep the VSM fixed throughout our experiments, changing only the context type. This methodology allows us to evaluate the impact of different contexts on the VSM performance, as context choice is the only modeling decision that changes across experimental conditions.

Our VSM is the word2vec skip-gram model (*w2v-SG*, Mikolov et al. (2013a)), which obtains state-of-the-art results on a variety of NLP tasks (Baroni et al., 2014). We employ the word2vec toolkit.[4] For all context types other than *BOW* we use the word2vec package of (Levy and Goldberg, 2014),[5] which augments the standard word2vec toolkit with code that allows arbitrary context definition.

Experimental Setup. We experiment with the verb pair (222 pairs) and adjective pair (111 pairs) portions of SimLex999 (Hill et al., 2014). We report the Spearman ρ correlation between the ranks derived from the scores of the evaluated models and the human scores provided in SimLex999.[6]

We train the *w2v-SG* model with five different context types: (a) *BOW* contexts (SG-*BOW*); (b) all dependency links (SG-*DepAll*) (c) dependency-based coordination contexts (i.e., those labeled with *conj*, SG-*Coor*); (d) all dependency links except for coordinations (SG-$Coor^C$); and (e) *SP* contexts. Our training corpus is the 8G words corpus generated by the word2vec script.[7]

Model	Verb	Adj.	Noun	Time	#Cont.
SG-*BOW*	0.307	0.604	**0.501**	320	13G
SG-*DepAll*	0.386	0.586	0.499	551	14.5G
SG-*Coor*	0.413	0.629	0.428	23	550M
SG-$Coor^C$	0.372	0.56	0.494	677	14G
SG-*SP*	**0.459**	**0.651**	0.415	11	270M
SRR15	0.578	0.663	0.497	—	270M
SRR15$^-$	0.441	0.68	0.421	—	270M

Table 1:
Spearman's ρ scores on the different portions of SimLex999. The top part presents results for the word2vec skip-gram model (*w2v-SG*) with various context types (see text). The bottom lines present the results of the count *SP*-based model of Schwartz et al. (2015), with (SRR15) and without (SRR15$^-$) its antonym detection method. The two rightmost columns present the run time of the *w2v-SG* models in minutes (Time) and the number of context instances used by the model (#Cont.).[10] For each SimLex999 portion, the score of the best *w2v-SG* model across context types is highlighted in bold font.

Models (b)-(d) require the dependency parse trees of the corpus as input. To generate these trees, we employ the Stanford POS Tagger (Toutanova et al., 2003)[8] and the stack version of the MALT parser (Nivre et al., 2009).[9] The *SP* contexts are generated using the *SPs* extracted by the DR06 algorithm from our training corpus (see Section 3).

For *BOW* contexts, we experiment with three window sizes (2, 5 and 10) and report the best results (window size of 2 across conditions). For dependency based contexts we follow the standard convention in the literature: we consider the immediate heads and modifiers of the represented word. All models are trained with 500 dimensions, the default value of the word2vec script. Other hyperparameters were also set to the default values of the code packages.

Results. Table 1 presents our results. The SG-*SP* model provides the most useful verb and adjective representations among the *w2v-SG* models. Compared to *BOW* (SG-*BOW*), the most commonly used

[4] `https://code.google.com/p/word2vec/`

[5] `https://bitbucket.org/yoavgo/word2vecf`

[6] Model scores are computed in the standard way: applying the cosine similarity metric to the vectors learned for the words participating in the pair.

[7] `code.google.com/p/word2vec/source/browse/trunk/demo-train-big-model-v1.sh`

[8] `nlp.stanford.edu/software/tagger.shtml`

[9] `http://www.maltparser.org/index.html`

context type, SG-*SP* results are 15.2% and 4.7% higher on verbs and adjectives respectively. Compared to dependency links (SG-*DepAll*), the improvements are 7.3% and 6.5%. For completeness, we compare the models on the noun pairs portion, observing that SG-*BOW* and SG-*DepAll* are ∼8.5% better than SG-*SP*. This indicates that different word classes require different representations.

The results for SG-*Coor*, which is trained with syntactic coordination (*Coor*) contexts, show that these contexts are superior to all the other dependency links (SG-$Coor^C$) by 4.1% and 6.9% on verbs and adjectives. Importantly, comparing the SG-*Coor* model to the SG-*DepAll* model, which augments the *Coor* contexts with the other syntactic dependency contexts, reveals that SG-*DepAll* is actually inferior by 2.7% and 4.3% in Spearman ρ on verbs and adjectives respectively. Interestingly, *Coor* contexts, which are extracted using a supervised parser, are still inferior by 4.6% and 2.2% to *SPs*, which capture similar contexts but are extracted from plain text.

Table 1 also shows the training times of the various *w2v-SG* models on a 32G memory, 32 CPU core machine. SG-*SP* and SG-*Coor*, which take 11 minutes and 23 minutes respectively to train, are substantially faster than the other *w2v-SG* models. For example, they are more than an order of magnitude faster than SG-*BOW* (320 minutes) and SG-$Coor^C$ (677 minutes). This is not surprising, as there are far fewer *SP* contexts (270M) and *Coor* contexts (550M) than *BOW* contexts (13G) and $Coor^C$ contexts (14G) (#Cont. column).

Finally, the performance of the SG-*SP* model is still substantially inferior to the SRR15 *SP*-based model (Schwartz et al., 2015). As both models use the same *SP* contexts, this result indicates that other modeling decisions in SRR15 lead to its superior performance. We show that this difference is mostly attributed to one feature of SRR15: its method for detecting antonym pairs (*good/bad*). Indeed, the SRR15 model without its antonym detection method (SRR15$^-$) obtains a Spearman ρ of 0.441, compared to 0.459 of SG-*SP* on verb pairs. For adjectives, however, SRR15$^-$ is 1.7% better than SRR15, increasing the difference from SG-*SP* to 2.9%.[11]

5 Conclusions

We demonstrated the effectiveness of symmetric pattern contexts in word embedding induction. Experiments with the word2vec model showed that these contexts are superior to various alternatives for verb and adjective representation. We further pointed at the connection between symmetric patterns and syntactic coordinations. We showed that coordinations are superior to other syntactic contexts, but are still inferior to symmetric patterns, although the extraction of symmetric patterns requires less supervision.

Future work includes developing a model that successfully combines the various context types explored in this paper. We are also interested in the representation of other word classes such as adverbs for which no evaluation set currently exists. Finally, the code for generating the SG-*SP* embeddings, as well as the vectors experimented with in this paper, are released and can be downloaded from `http://www.cs.huji.ac.il/~roys02/papers/sp_sg/sp_sg.html`

Acknowledgments

This research was funded (in part) by the Intel Collaborative Research Institute for Computational Intelligence (ICRI-CI), the Israel Ministry of Science and Technology Center of Knowledge in Machine Learning and Artificial Intelligence (Grant number 3-9243). The second author was partially funded by the Microsoft/Technion research center for electronic commerce and the Google faculty research award.

References

Simon Baker, Roi Reichart, and Anna Korhonen. 2014. An unsupervised model for instance level subcategorization acquisition. In *Proc. of EMNLP*.

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proc. of ACL*.

[10] We compare the *w2v-SG* models training time only. SRR15 and SRR15$^-$ are count-based models and have no training step.

[11] We report results for our reimplementation of SRR15 and SRR15$^-$.

Elia Bruni, Nam-Khanh Tran, and Marco Baroni. 2014. Multimodal distributional semantics. *JAIR*.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *JMLR*, 12:2493–2537.

Dmitry Davidov and Ari Rappoport. 2006. Efficient unsupervised discovery of word categories using symmetric patterns and high frequency words. In *Proc. of ACL-COLING*.

Dmitry Davidov, Ari Rappoport, and Moshe Koppel. 2007. Fully unsupervised discovery of concept-specific relationships by web mining. In *Proc. of ACL*.

Dmitry Davidov, Oren Tsur, and Ari Rappoport. 2010. Enhanced sentiment learning using twitter hashtags and smileys. In *Proc. of COLING*.

Beate Dorow, Dominic Widdows, Katarina Ling, Jean-Pierre Eckmann, Danilo Sergi, and Elisha Moses. 2005. Using Curvature and Markov Clustering in Graphs for Lexical Acquisition and Word Sense Discrimination.

Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S Weld, and Alexander Yates. 2005. Unsupervised named-entity extraction from the web: An experimental study. *Artificial intelligence*, 165(1):91–134.

Song Feng, Jun Seok Kang, Polina Kuznetsova, and Yejin Choi. 2013. Connotation lexicon: A dash of sentiment beneath the surface meaning. In *Proc. of ACL*.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2001. Placing search in context: The concept revisited. In *Proc. of WWW*.

Gregory Grefenstette. 1994. *Explorations in automatic thesaurus discovery*. Boston: Kluwer.

Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proc. of COLING – Volume 2*.

Felix Hill, Roi Reichart, and Anna Korhonen. 2014. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *arXiv:1408.3456 [cs.CL]*.

Richard A. Hudson. 1980. *Arguments for a Non-transformational Grammar*. Chicago: University of Chicago Press.

Zornitsa Kozareva, Ellen Riloff, and Eduard Hovy. 2008. Semantic class learning from the web with hyponym pattern linkage graphs. In *Proc. of ACL-HLT*.

Omer Levy and Yoav Goldberg. 2014. Dependency-based word embeddings. In *Proc. of ACL*.

Omer Levy, Steffen Remus, Chris Biemann, and Ido Dagan. 2015. Do supervised distributional methods really learn lexical inference relations? In *Proc. of NAACL*.

Dekang Lin, Shaojun Zhao, Lijuan Qin, and Ming Zhou. 2003. Identifying synonyms among distributionally similar words. In *Proc. of IJCAI*.

Oren Melamud, David McClosky, Siddharth Patwardhan, and Mohit Bansal. 2016. The role of context types and dimensionality in learning word embeddings. In *Proc. of NAACL*.

Igor Mel'čuk. 1988. *Dependency Syntax: Theory and Practice*. State University of New York Press.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013a. Distributed representations of words and phrases and their compositionality. In *Proc. of NIPS*.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013b. Linguistic regularities in continuous space word representations. In *Proc. of NAACL-HLT*.

George A Miller and Walter G Charles. 1991. Contextual correlates of semantic similarity. *Language and cognitive processes*.

Preslav Nakov and Marti Hearst. 2005. Using the web as an implicit training set: application to structural ambiguity resolution. In *Proc. of HLT-EMNLP*.

Jens Nilsson, Joakim Nivre, and Johan Hall. 2006. Graph transformations in data-driven dependency parsing. In *Proc. of ACL-COLING*.

Joakim Nivre, Marco Kuhlmann, and Johan Hall. 2009. An improved oracle for dependency parsing with online reordering. In *Proc. of IWPT*.

Sebastian Padó and Mirella Lapata. 2007. Dependency-based construction of semantic space models. *Computational Linguistics*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proc. of EMNLP*.

Herbert Rubenstein and John B. Goodenough. 1965. Contextual correlates of synonymy. *Communications of the ACM*.

Dana Rubinstein, Effi Levi, Roy Schwartz, and Ari Rappoport. 2015. How well do distributional models capture different types of semantic knowledge? In *Proc. of ACL*.

Roy Schwartz, Omri Abend, Roi Reichart, and Ari Rappoport. 2011. Neutralizing linguistically problematic annotations in unsupervised dependency parsing evaluation. In *Proc. of ACL-HLT*.

Roy Schwartz, Omri Abend, and Ari Rappoport. 2012. Learnability-based syntactic annotation design. In *Proc. of COLING*.

Roy Schwartz, Oren Tsur, Ari Rappoport, and Moshe Koppel. 2013. Authorship attribution of micro-messages. In *Proc. of EMNLP*.

Roy Schwartz, Roi Reichart, and Ari Rappoport. 2014. Minimally supervised classification to semantic categories using automatically acquired symmetric patterns. In *Proc. of COLING*.

Roy Schwartz, Roi Reichart, and Ari Rappoport. 2015. Symmetric pattern based word embeddings for improved word similarity prediction. In *Proc. of CoNLL*.

Lucien Tesnière. 1959. *Éléments de syntaxe structurale*. Paris: K1incksieck.

Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proc. of NAACL*.

Peter D. Turney. 2006. Similarity of semantic relations. *Computational Linguistics*.

Peter D. Turney. 2008. A uniform approach to analogies, synonyms, antonyms, and associations. In *Proc. of COLING*.

Dominic Widdows and Beate Dorow. 2002. A graph model for unsupervised lexical acquisition. In *Proc. of COLING*.

Dongqiang Yang and David M. W. Powers. 2006. Verb similarity on the taxonomy of wordnet. In *Proc. of GWC*.

Published in EMNLP 2013

Chapter 5

Authorship Attribution of Micro-Messages

Authorship Attribution of Micro Messages

Roy Schwartz, Oren Tsur, Ari Rappoport
Institute of Computer Science
Hebrew University of Jerusalem
{roys02|oren|arir}@cs.huji.ac.il

Moshe Koppel
Department of Computer Science
Bar Ilan University
koppel@macs.biu.ac.il

Abstract

Work on authorship attribution has traditionally focused on long texts. In this work, we tackle the question of whether the author of a very short text can be successfully identified. We use Twitter as an experimental testbed. We introduce the concept of an author's unique "signature", and show that such signatures are typical of many authors when writing very short texts. We also present a new authorship attribution feature ("flexible patterns") and demonstrate a significant improvement over our baselines. Our results show that the author of a single tweet can be identified with good accuracy in an array of flavors of the authorship attribution task.

1 Introduction

Research in authorship attribution has developed substantially over the last decade (Stamatatos, 2009). The vast majority of such research has been dedicated towards finding the author of long texts, ranging from single passages to book chapters. In recent years, the growing popularity of social media has created special interest, both theoretical and computational, in short texts. This has led to many recent authorship attribution projects that experimented with web data such as emails (Abbasi and Chen, 2008), web forum messages (Solorio et al., 2011) and blogs (Koppel et al., 2011b). This paper addresses the question to what extent the authors of very short texts can be identified. To answer this question, we experiment with Twitter tweets.

Twitter messages (tweets) are limited to 140 characters. This restriction imposes major difficulties on authorship attribution systems, since authorship attribution methods that work well on long texts are often not as useful when applied to short texts (Burrows, 2002; Sanderson and Guenter, 2006).

Nonetheless, tweets are relatively self-contained and have smaller sentence length variance compared to excerpts from longer texts (see Section 3). These characteristics make Twitter data appealing as a testbed when focusing on short texts. Moreover, an authorship attribution system of tweets may have various applications. Specifically, a range of cybercrimes can be addressed using such a system, including identity fraud and phishing.

In this paper, we introduce the concept of k-*signatures*. We denote the k-*signatures* of an author a as the features that appear in at least k% of a's training samples, while not appearing in the training set of any other author. When k is large, such signatures capture a unique style used by a. An analysis of our training set reveals that unique k-signatures are typical of many authors. Moreover, a substantial portion of the tweets in our training set contain at least one such signature. These findings suggest that a single tweet, although short and sparse, often contains sufficient information for identifying its author. Our results show that this is indeed the case.

We train an SVM classifier with a set of features that include character n-grams and word n-grams. We use a rigorous experimental setup, with varying number of authors (values between 50-1,000) and various sizes of the training set, ranging from 50 to 1,000 tweets per author. In all our experiments, a single tweet is used as test document. We also use a setting in which the system is allowed to respond *don't know* in cases of uncertainty. Applying this option results in higher precision, at the expense of lower recall.

Our results show that the author of a tweet can be successfully identified. For example, when using a dataset of as many as 1,000 authors with 200 training tweets per author, we are able to obtain 30.3% accuracy (as opposed to a random baseline of only 0.1%). Using a dataset of 50 authors with as few as 50 training tweets per author, we obtain 50.7% accuracy. Using a dataset of 50 authors with 1,000 training tweets per author, our results reach as high as 71.2% in the standard classification setting, and exceed 91% accuracy with 60% recall in the *don't know* setting.

We also apply a new set of features, never previously used for this task – flexible patterns. Flexible patterns essentially capture the context in which function words are used. The effectiveness of function words as authorship attribution features (Koppel et al., 2009) suggests using flexible pattern features. The fact that flexible patterns are learned from plain text in a fully unsupervised manner makes them domain and language independent. We demonstrate that using flexible patterns gives significant improvement over our baseline system. Furthermore, using flexible patterns, our system obtains a 6.1% improvement over current state-of-the-art results in authorship attribution on Twitter.

To summarize, the contribution of this paper is threefold.

- We provide the most extensive research to date on authorship attribution of micro-messages, and show that authors of very short texts can be successfully identified.

- We introduce the concept of an author's unique k-signature, and demonstrate that such signatures are used by many authors in their writing of micro-messages.

- We present a new feature for authorship attribution – flexible patterns – and show its significant added value over other methods. Using this feature, our system obtains a 6.1% improvement over the current state-of-the-art.

The rest of the paper is organized as follows. Sections 2 and 3 describe our methods and our experimental testbed (Twitter). Section 4 presents the concept of k-signatures. Sections 5 and 6 present our experiments and results. Flexible patterns are presented in Section 7 and related work is presented in Section 8.

2 Methodology

In the following we briefly describe the main features employed by our system. The features below are binary features.

Character n-grams. Character n-gram features are especially useful for authorship attribution on micro-messages since they are relatively tolerant to typos and non-standard use of punctuation (Stamatatos, 2009). These are common in the non-formal style generally applied in social media services. Consider the example of misspelling "Britney" as "Brit**t**ney". The misspelled name shares the 4-grams "Brit" and "tney" with the correct name. As a result, these features provide information about the author's style (or at least her topic of interest), which is not available through lexical features.

Following standard practice, we use 4-grams (Sanderson and Guenter, 2006; Layton et al., 2010; Koppel et al., 2011b). White spaces are considered characters (i.e., a character n-gram may be composed of letters from two different words). A single white-space is appended to the beginning and the end of each tweet. For efficiency, we consider only character n-gram features that appear at least t_{cng} times in the training set of at least one author (see Section 5).

Word n-grams. We hypothesize that word n-gram features would be useful for authorship attribution on micro-messages. We assume that under a strict length restriction, many authors would prefer using short, repeating phrases (word n-grams).

In our experiments, we consider $2 \leq n \leq 5$.[1] We regard sequences of punctuation marks as words. Two special words are added to each tweet to indicate the beginning and the end of the tweet. For efficiency, we consider only word n-gram features that appear at least t_{wng} times in the training set of at least one author (see Section 5).

Model. We use libsvm's Matlab implementation of a multi-class SVM classifier with a linear kernel

[1] We skip unigrams as they are generally captured by the character n-gram features.

(Chang and Lin, 2011). We use ten-fold cross validation on the training set to select the best regularization factor between 0.5 and 0.005.[2]

3 Experimental Testbed

Our main research question in this paper is to determine the extent to which authors of very short texts can be identified. A major issue in working with short texts is selecting the right dataset. One approach is breaking longer texts into shorter chunks (Sanderson and Guenter, 2006). We take a different approach and experiment with micro-messages (specifically, tweets).

Tweets have several properties making them an ideal testbed for authorship attribution of short texts. First, tweets are posted as single units and do not necessarily refer to each other. As a result, they tend to be self contained. Second, tweets have more standardized length distribution compared to other types of web data. We compared the mean and standard deviation of sentence length in our Twitter dataset and in a corpus of English web data (Ferraresi et al., 2008).[3] We found that (a) tweets are shorter than standard web data (14.2 words compared to 20.9), and (b) the standard deviation of the length of tweets is much smaller (6.4 vs. 21.4).

Pre-Processing. We use a Twitter corpus that includes approximately 5×10^8 tweets.[4] All non-English tweets and tweets that contain fewer than 3 words are removed from the dataset. We also remove tweets marked as retweets (using the RT sign, a standard Twitter symbol to indicate that this tweet was written by a different user). As some users retweet without using the RT sign, we also remove tweets that are an exact copy of an existing tweet posted in the previous seven days.

Apart from plain text, some tweets contain references to other Twitter users (in the format of @<user>). Since using reference information makes this task substantially easier (Layton et al., 2010), we replace each user reference with the special meta tag REF. For sparsity reasons, we also replace web addresses with the meta tag URL, numbers with the meta tag NUM, time of day with the meta tag TIME and dates with the meta tag DATE.

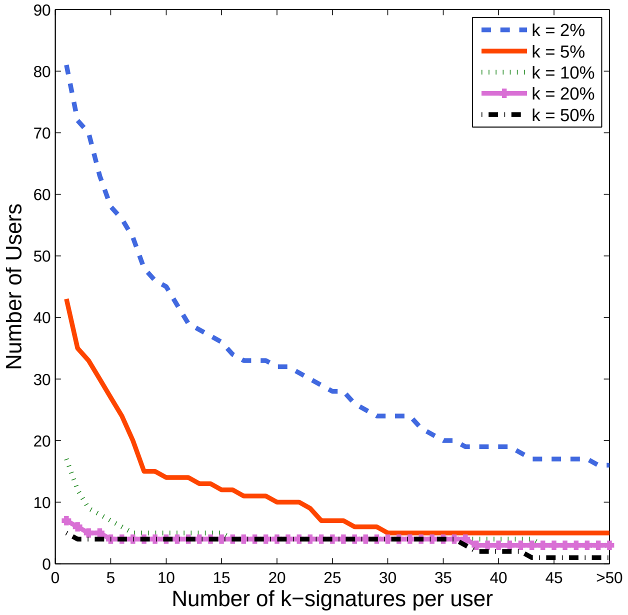


Figure 1: Number of users with at least x k-signatures (100 authors, 180 training tweets per author).

4 k-Signatures

In this section, we show that many authors adopt a unique style when writing micro-messages. This style can be detected by a strong classification algorithm (such as SVM), and be sufficient to correctly identify the author of a single tweet.

We define the concept of the k-*signature* of an author a to be a feature that appears in at least k% of a's training set, while not appearing in the training set of any other user. Such signatures can be useful for identifying future (unlabeled) tweets written by a.

To validate our hypothesis, we use a dataset of 100 authors with 180 tweets per author. We compute the number of k-signatures used by each of the authors in our dataset. Figure 1 shows our results for a range of k values (2%, 5%, 10%, 20% and 50%). Results demonstrate that 81 users use at least one 2%-signature, 43 users use at least one 5%-signature, and 17 users use at least one 10%-signature. These results indicate that a large portion of the users adopt a unique signature (or set of signatures) when writing short texts. Table 1 provides examples of 10%-signatures.

[2] In practice, 0.05 or 0.1 are selected in almost all cases.

[3] `http://wacky.sslmit.unibo.it`

[4] These comprise ∼15% of all public tweets created from May 2009 to March 2010.

Signature Type	10%-signature	Examples
Character n-grams	‘ ^_^’	REF oh ok ^_^ Glad you found it!
		Hope everyone is having a good afternoon ^_^
		REF Smirnoff lol keeping the goose in the freezer ^_^
	‘**yew** ’	gurl **yew** serving me tea nooch
		REF about wen **yew** and ronnie see each other
		REF lol so **yew** goin to check out tini’s tonight huh???
Word n-grams	**.. lal**	REF aww those are cool where u get those.. how do ppl react**.. lal**
		Ludas album is gone be hott**.. lal**
		Dayum refs don’t get injury timeouts**.. lal**.. get him off the field..
	smoochies , e3	I’m just back after takin’ a very long, icy cold shower........Shivering **smoochies,E3** http://bit.ly/4CzzP9
		A blue stout or two would be nice as well, Purr!Blue smooth **smoochies,E3** http://bit.ly/75D4fO
		That is sooooooooooooooooooo unfair!Double **smoochies,E3** http://bit.ly/07sXRGX

Table 1: Examples of 10%-signatures.

Results also show that seven users use one or more 20%-signatures, and five users even use one or more 50%-signatures. Looking carefully at these users, we find that they write very structured messages, and are probably bots, such as news feeds, bidding systems, etc. Table 2 provides examples of tweets posted by such users.[5]

Another interesting question is how many tweets contain at least one k-signature. Figure 2 shows for each user the number of tweets in her training set for which at least one k-signature is found. Results demonstrate that a total of 18.6% of the training tweets contain at least one 2%-signature, 10.3% the training tweets contain at least one 5%-signature and 6.5% of the training tweets contain at least one 10%-signature. These findings validate our assumption that many users use k-signatures in short texts.

These findings also have direct implications on authorship attribution of micro-messages, since k-signatures are reliable classification features. As a result, texts written by authors that tend to use k-signatures are likely to be easily identified by a reasonable classification algorithm. Consequently, k-signatures provide a possible explanation for the high quality results presented in this paper.

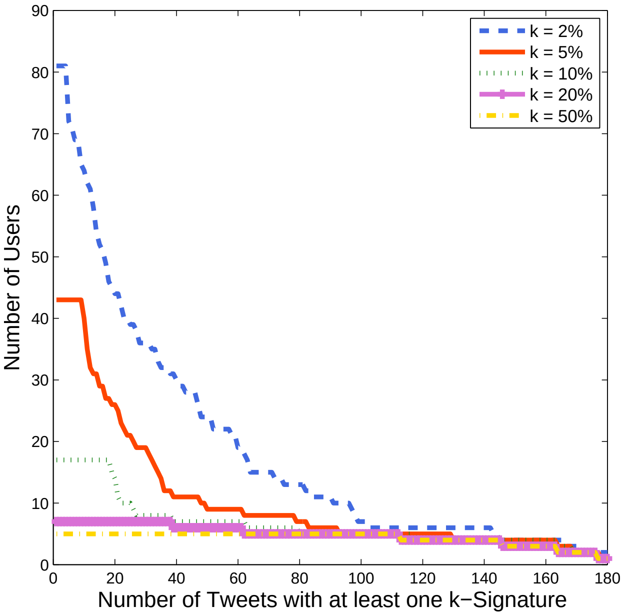


Figure 2: Number of users with at least x training tweets that contain at least one k-signature (100 authors, 180 training tweets per author).

In the broader context, the presence (and contribution) of k-signatures is in line with the hypothesis proposed by (Davidov et al., 2010a): while still using an informal and unstructured (grammatical) language, authors tend to use typical and unique structures in order to allow a short message to stand alone without a clear conversational context.

[5] Our k-signature method can actually be useful for automatically identifying such users. We defer this to future work.

User	20%-signature	Examples
1	**I'm listening to :**	**I'm listening to:** Sigur Rós ? Intro: http://www.last.fm/music/Sigur+Rós http://bit.ly/3XJHyb
		I'm listening to: Tina Arena ? In Command: http://www.last.fm/music/Tina+Arena http://bit.ly/7q9E25
		I'm listening to: Midnight Oil ? Under the Overpass: http://www.last.fm/music/Midnight+Oil http://bit.ly/7IH4cg
2	**news now (str)**	#Hotel **News Now(STR)** 5 things to know: 27 May 2009: From the desks of the HotelNewsNow.com editor... http://bit.ly/aZTZOq #Tourism #Lodging
		#Hotel **News Now(STR)** Five sales renegotiating tactics: As bookings representatives press to reneg... http://bit.ly/bHPn2L
		#Hotel **News Now(STR)** Risk of hotel recession retreats: The Hotel Industry's Pulse Index increases... http://bit.ly/a8EKrm #Tourism #Lodging
3	**(NUM bids) end date :**	NEW PINK NINTENDO DS LITE CONSOLE WITH 21 GIFTS + CASE: &#163;66.50 **(13 Bids) End Date:** Tuesday Dec-08-2009 17:.. http://bit.ly/7uPt6V
		Microsoft Xbox 360 Game System - Console Only - Working: US $51.99 **(25 Bids) End Date:** Saturday Dec-12-2009 13:.. http://bit.ly/8VgdTv
		Microsoft Sony Playstation 3 (80 GB) Console 6 Months Old: &#163;190.00 **(25 Bids) End Date:** Sunday Dec-13-2009 21:21:39 G.. http://bit.ly/7kwtDS

Table 2: Examples of tweets published by very structured users, suspected to be bots, along with one of their 20%-signatures.

5 Experiments

We report of three different experimental configurations. In the experiments described below, each dataset is divided into training and test sets using ten-fold cross validation. On the test phase, each document contains a single tweet.

Experimenting with varying Training Set Sizes. In order to test the affect of the training set size, we experiment with an increasingly larger number of tweets per author. Experimenting with a range of training set sizes serves two purposes: (a) to check whether the author of a tweet can be identified using a very small number of (short) training samples, and (b) check how much our system can benefit from training on a larger corpus.

In our experiments we only consider users who posted between 1,000–2,000 tweets[6] (a total of 10,183 users), and randomly select 1,000 tweets per user. From these users, we select 10 groups of 50 users each.[7] We perform a set of classification experiments, selecting for each author an increasingly larger subset of her 1,000 tweets as training set. Subset sizes are (50, 100, 200, 500, 1,000). Threshold values for our features in each setting (see Section 2) are $(2, 2, 4, 10, 20)$ for t_{cng} and $(2, 2, 2, 3, 5)$ for t_{wng}, respectively.

Experimenting with varying Numbers of Authors. In a second set of experiments, we use an increasingly larger number of authors (values between 100-1,000), in order to check whether the author of a very short text can be identified in a "needle in a haystack" type of setting.

Due to complexity issues, we only experiment with 200 tweets per author as training set. We select groups of size 100, 200, 500 and 1,000 users (one group per size). We use the same threshold values as the 200 tweets per author setting previously described (t_{cng} = 4, t_{wng} = 2).

[6] This range is selected since on one hand we want at least 1,000 tweets per author for our experiments, and on the other hand we noticed that users with a larger number of tweets in corpus tend to be spammers or bots that are very easy to identify, so we limit this number to 2,000.

[7] An eleventh group is selected as development set.

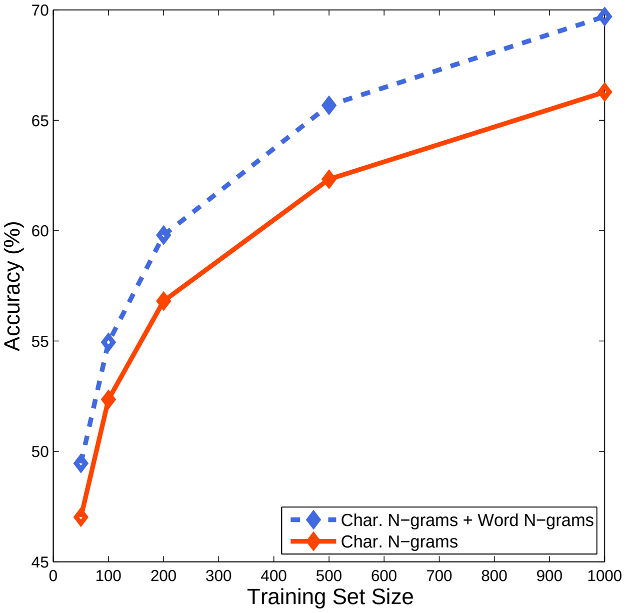


Figure 3: Authorship attribution accuracy for 50 authors with various training set sizes. The values are averaged over 10 groups. The random baseline is 2%.

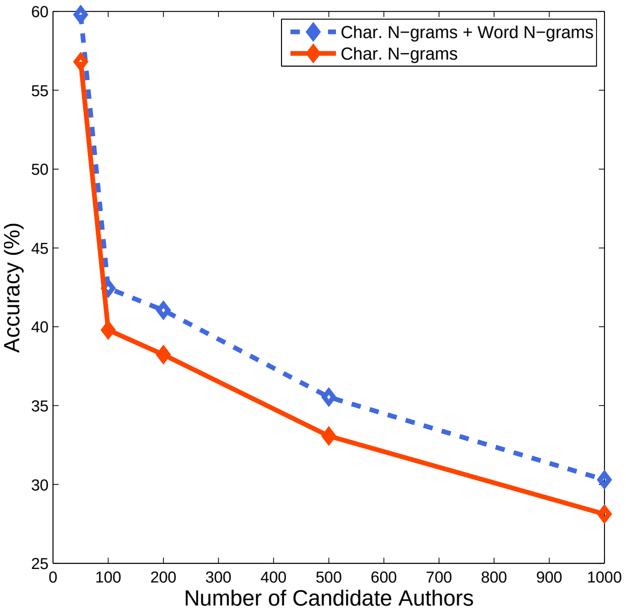


Figure 4: Authorship attribution accuracy with varying number of candidate authors, using 200 training tweets per author. The random baselines for 50[9], 100, 200, 500 and 1,000 authors are 2%, 1%, 0.5%, 0.2% and 0.1%, respectively.

Recall-Precision Tradeoff. Another aspect of our research question is the level of certainty our system has when suggesting an author for a given tweet. In cases of uncertainty, many real life applications would prefer not to get any response instead of getting a response with low certainty. Moreover, in real life applications we are often not even sure that the real author is part of our training set. Consequently, we allow our system to respond "*don't know*" in cases of low confidence (Koppel et al., 2006; Koppel et al., 2011b). This allows our system to obtain higher precision, at the expense of lower recall.

To implement this feature, we use SVM's probability estimates, as implemented in libsvm. These estimates give a score to each potential author. These scores reflect the probability that this author is the correct author, as decided by the prediction model. The selected author is always the one with the highest probability estimate.

As selection criterion, we use a set of increasingly larger thresholds (0.05-0.9) for the probability of the selected author. This means that we do not select test samples for which the selected author has a probability estimate value lower than the threshold.

6 Basic Results

Experimenting with varying Training Set Sizes. Figure 3 shows results for our experiments with 50 authors and various training set sizes. Results demonstrate that authors of very short texts can be successfully identified, even with as few as 50 tweets per author (49.5%). When given more training samples, authors are identified much more accurately (up to 69.7%). Results also show that, according to our hypothesis, word n-gram features substantially improve over character n-grams features only (3% averaged improvement over all settings).

Experimenting with varying Numbers of Authors. Figure 4 shows our results for various numbers of authors, using 200 tweets per author as training set. Results demonstrate that authors of an unknown tweet can be identified to a large extent even when there are as many as 1,000 candidate authors (30.3%, as opposed to a random baseline of only 0.1%). Results further validate that word n-gram features substantially improve over character

[9] Results for 50 authors with 200 tweets per author are taken from Figure 3.

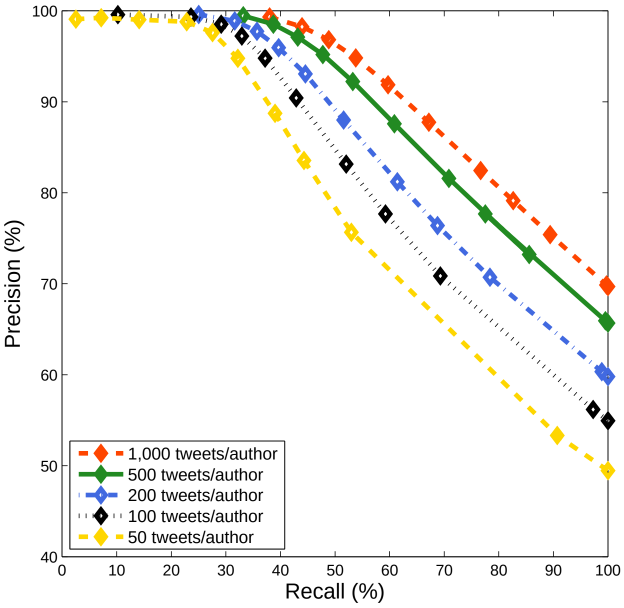


Figure 5: Recall-precision curves for 50 authors with varying training set sizes.

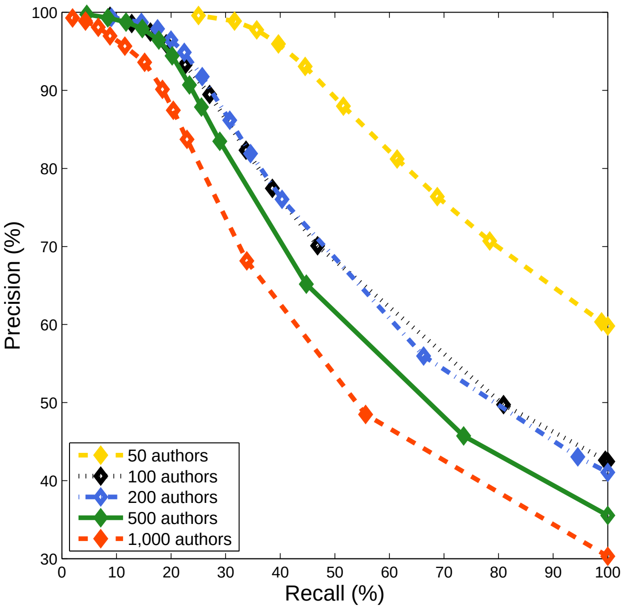


Figure 6: Recall-precision curves for varying number of authors.

n-grams features (2.6% averaged improvement).

Recall-Precision Tradeoff. Figure 5 shows the recall-precision curves for our experiments with 50 authors and varying training set sizes. Results demonstrate that we are able to obtain very high precision (over 90%) while still maintaining a relatively high recall (from ∼35% recall for 50 tweets per author up to $> 60\%$ recall for 1,000 tweets per author).

Figure 6 shows the recall-precision curves for our experiments with varying number of authors. Results demonstrate that even in the 1,000 authors setting, we are able to obtain high precision values (90% and 70%) with reasonable recall values (18% and ∼30%, respectively).

7 Flexible Patterns

In previous sections we provided strong evidence that authors of micro-messages can be successfully identified using standard methods. In this section we present a new feature, never previously used for this task – flexible patterns. We show that flexible patterns can be used to improve classification results.

Flexible patterns are a generalization of word n-grams, in the sense that they capture potentially unseen word n-grams. As a result, flexible patterns can pick up fine-grained differences between authors' styles. Unlike other types of pattern features, flexible patterns are computed automatically from plain text. As such, they can be applied to various tasks, independently of domain and language. We describe them in detail.

Word Frequency. Flexible patterns are composed of high frequency words (HFW) and content words (CW). Every word in the corpus is defined as either HFW or CW. This clustering is performed by counting the number of times each word appears in the corpus of size s. A word that appears more than $10^{-4} \times s$ times in a corpus is considered HFW. A word that appears less than $10^{-3} \times s$ times in a corpus is considered CW. Some words may serve both as HFWs and CWs (see Davidov and Rappoport (2008b) for discussion).

Structure of a Flexible Pattern. Flexible patterns start and end with an HFW. A sequence of zero or more CWs separates consecutive HFWs. At least one CW must appear in every pattern.[10] For efficiency, at most six HFWs (and as a result, five CW sequences) may appear in a flexible pattern. Examples of flexible patterns include

1. “*the*$_{HFW}$ *CW* *of*$_{HFW}$ *the*$_{HFW}$”

[10] Omitting this treats word n-grams as flexible patterns.

Flexible Pattern Features. Flexible patterns can serve as binary classification features; a tweet matches a given flexible pattern if it contains the flexible pattern sequence. For example, (1) is matched by (2).

2. "*Go to* <u>*the*$_{HFW}$ *house*$_{CW}$ *of*$_{HFW}$ *the*$_{HFW}$</u> *rising sun*"

Partial Flexible Patterns. A flexible pattern may appear in a given tweet with additional words not originally found in the flexible pattern, and/or with only a subset of the HFWs (Davidov et al., 2010a). For example, (3) is a partial match of (1), since the word "great" is not part of the original flexible pattern. Similarly, (4) is another partial match of (1), since (a) the word "good" is not part of the original flexible pattern and (b) the second occurrence of the word "the" does not appear in (4) (missing word is marked by ___).

3. "<u>*The*$_{HFW}$ **great**$_{HFW}$ *king*$_{CW}$ *of*$_{HFW}$ *the*$_{HFW}$</u> *ring*"

4. "<u>*The*$_{HFW}$ **good**$_{HFW}$ *king*$_{CW}$ *of*$_{HFW}$ ___</u> *Spain*"

We use such cases as features with lower weight, proportional to the number of found HFWs in the tweet ($w = \frac{0.5 \times n_{found}}{n_{expected}}$). For example, (1) receives a weight of 1 (complete match) against (2). Against (3), it receives a weight of 0.5 ($= \frac{0.5 \times 3}{3}$, partial match with no missing HFWs). Against (4) it receives a weight of 1/3 ($= \frac{0.5 \times 2}{3}$, partial match with only 2/3 HFWs found).

Experimenting with Flexible Pattern Features. We repeat our experiments with varying training set sizes (see Section 5) with two more systems: one that uses character n-grams and flexible pattern features, and another that uses character n-grams, word n-grams and flexible patterns. High frequency word counts are computed separately for each author using her training set. We only consider flexible pattern features that appear at least t_{fp} times in the training set of at least one author. Values of t_{fp} for training set sizes (50, 100, 200, 500, 1,000) are (2, 3, 7, 7, 8), respectively.

Results. Figure 7 shows our results. Results demonstrate that flexible pattern features have an added value over both character n-grams alone (averaged 2.9% improvement) and over character n-grams and word n-grams together (averaged 1.5% improvement). We perform t-tests on each of our training set sizes to check whether the latter improvement is significant. Results demonstrate that it is highly significant in all settings, with p-values smaller than values between 10^{-3} (for 50 tweets per author) and 10^{-8} (1,000 tweets per author).

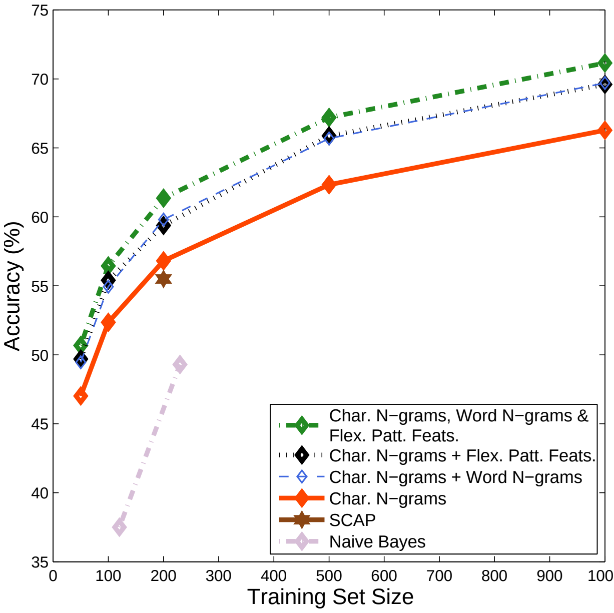


Figure 7: Authorship attribution accuracy for 50 authors with various training set sizes and various feature sets. The values are averaged over 10 groups. The random baseline is 2%.
Comparison to previous work: SCAP – SCAP algorithm results, as reported by (Layton et al., 2010), Naive Bayes – Naive Bayes algorithm results, as reported by (Boutwell, 2011).

Comparison to Previous Works. Figure 7 also shows results for the only two works that experimented in some of the settings we experimented in: Layton et al. (2010) and Boutwell (2011) (see Section 8). Our system substantially outperforms these two systems, by margins of 5.9% to 19%. These margins are explained by the choice of algorithm (SVM and not SCAP/naive Bayes) and our set of features (character n-grams + word n-grams + flexible patterns compared to character n-grams only). In order to rule out the possibility that these margins stem from using different datasets, we tested our system on the dataset used in (Layton et al., 2010). Our system obtains even higher results on this dataset than on our datasets (61.6%, a total improvement of 6.1% over (Layton et al., 2010)).

Discussion. To illustrate the additional contribution of flexible patterns over word n-grams, consider the following tweets, written by the same author.

5. "… $\underline{\textit{the}_{HFW}\ \textit{way}_{CW}\ \textit{I}_{HFW}}$ $\textit{treated}_{CW}$ $\textit{her}_{HFW}$"

6. "… *half of* $\underline{\textit{the}_{HFW}\ \textit{things}_{CW}\ \textit{I}_{HFW}}$ *have seen*"

7. "… $\underline{\textit{the}_{HFW}\ \textit{friends}_{CW}\ \textit{I}_{HFW}}$ *have had for years"*

8. "… *in* $\underline{\textit{the}_{HFW}\ \textit{neighborhood}_{CW}\ \textit{I}_{HFW}}$ *grew up in*"

Consider a case where (5) is part of the test set, while (6-8) appear in the training set. As (5) shares no sequence of words with (6-8), no word n-gram feature is able to identify the author's style in (5). However, this style can be successfully identified using the flexible pattern (9), shared by (5-8).

9. $\textit{the}_{HFW}\quad \textit{CW}\quad \textit{I}_{HFW}$

This demonstrates the added value flexible pattern features have over word n-gram features.

8 Related Work

Authorship attribution dates back to the end of 19th century, when (Mendenhall, 1887) applied sentence length and word length features to plays of Shakespeare. Ever since, many methods have been developed for this task. For recent surveys, see (Koppel et al., 2009; Stamatatos, 2009; Juola, 2012).

Authorship attribution methods can be generally divided into two categories (Stamatatos, 2009). In similarity-based methods, an anonymous text is attributed to some author whose writing style is most similar (by some distance metric). In machine learning methods, which we follow in this paper, anonymous texts are classified, using machine learning algorithms, into different categories (in this case, different authors).

Machine learning papers differ from each other by the features and machine learning algorithm. Examples of features include HFWs (Mosteller and Wallace, 1964; Argamon et al., 2007), character n-gram (Kjell, 1994; Hoorn et al., 1999; Stamatatos, 2008), word n-grams (Peng et al., 2004), part-of-speech n-grams (Koppel and Schler, 2003; Koppel et al., 2005) and vocabulary richness (Abbasi and Chen, 2005).

The various machine learning algorithms used include naive Bayes (Mosteller and Wallace, 1964; Kjell, 1994), neural networks (Matthews and Merriam, 1993; Kjell, 1994), K-nearest neighbors (Kjell et al., 1995; Hoorn et al., 1999) and SVM (De Vel et al., 2001; Diederich et al., 2003; Koppel and Schler, 2003).

Traditionally, authorship attribution systems have mainly been evaluated against long texts such as theater plays (Mendenhall, 1887), essays (Yule, 1939; Mosteller and Wallace, 1964), biblical books (Mealand, 1995; Koppel et al., 2011a) and book chapters (Argamon et al., 2007; Koppel et al., 2007). In recent year, many works focused on web data such as emails (De Vel et al., 2001; Koppel and Schler, 2003; Abbasi and Chen, 2008), web forum messages (Abbasi and Chen, 2005; Solorio et al., 2011), blogs (Koppel et al., 2006; Koppel et al., 2011b) and chat messages (Abbasi and Chen, 2008). Some works focused on SMS messages (Mohan et al., 2010; Ishihara, 2011).

Authorship Attribution on Twitter. The performance of authorship attribution systems on short texts is affected by several factors (Stamatatos, 2009). These factors include the number of candidate authors, the training set size and the size of the test document.

Very few authorship attribution works experimented with Twitter. Unlike our work, all used a single group of authors (group sizes varied between 3-50). Layton et al. (2010) used the SCAP methodology (Frantzeskou et al., 2007) with character n-gram features. They experimented with 50 authors and compared different numbers of tweets per author (values between 20-200). Surprisingly, they showed that their system does not improve when given more training tweets. In our work, we noticed a different trend, and showed that more data can be extremely valuable for authorship attribution systems on micro-messages (see Section 6). Silva et al. (2011) trained an SVM classifier with various features (e.g., punctuation and vocabulary features) on a small dataset of three authors only, with varying training set size. Although their work used a set of Twitter-specific features that we do not explicitly use, our features implicitly cover a large portion of their features (such as punctuation and emoticon features, which are largely covered by character n-grams).

Boutwell (2011) used a naive Bayes classifier with character n-gram features. She experimented with 50 authors and two training size values (120 and 230). She also provided a set of experiments that studied the effect of joining several tweets into a single document. Mikros and Perifanos (2013) trained an SVM classifier with character n-gram and word n-grams. They experimented with 10 authors of Greek text, and also joined several tweets into a single document. Joining several tweets into a longer document is appealing since it can lead to substantial improvement of the classification results, as demonstrated by the works above. However, this approach requires the test data to contain several tweets that are known a-priori to be written by the same author. This assumption is not always realistic. In our paper, we intentionally focus on a single tweet as document size.

Flexible Patterns. Patterns were introduced by (Hearst, 1992), who used hand crafted patterns to discover hyponyms. Hard coded patterns were used for many tasks, such as discovering meronymy (Berland and Charniak, 1999), noun categories (Widdows and Dorow, 2002), verb relations (Chklovski and Pantel, 2004) and semantic class learning (Kozareva et al., 2008).

Patterns were first extracted in a fully unsupervised manner ("flexible patterns") by (Davidov and Rappoport, 2006), who used flexible patterns in order to establish noun categories, and (Biciçi and Yuret, 2006) who used them for analogy question answering. Ever since, flexible patterns were used as features for various tasks such as extraction of semantic relationships (Davidov et al., 2007; Turney, 2008b; Bollegala et al., 2009), detection of synonyms (Turney, 2008a), disambiguation of nominal compound relations (Davidov and Rappoport, 2008a), sentiment analysis (Davidov et al., 2010b) and detection of sarcasm (Tsur et al., 2010).

9 Conclusion

The main goal of this paper is to measure to what extent authors of micro-messages can be identified. We have shown that authors of very short texts can be successfully identified in an array of authorship attribution settings reported for long documents. This is the first work on micro-messages to address some of these settings. We introduced the concept of k-*signature*. Using this concept, we proposed an interpretation of our results. Last, we presented the first authorship attribution system that uses flexible patterns, and demonstrated that using these features significantly improves over other systems. Our system obtains 6.1% improvement over the current state-of-the-art.

Acknowledgments

We would like to thank Elad Eban and Susan Goodman for their helpful advice, as well as Robert Layton for providing us with his dataset. This research was funded (in part) by the Harry and Sylvia Hoffman leadership and responsibility program (for the first author) and the Intel Collaborative Research Institute for Computational Intelligence (ICRI-CI).

References

Ahmed Abbasi and Hsinchun Chen. 2005. Applying authorship analysis to extremist-group web forum messages. *IEEE Intelligent Systems*, 20:67–75.

Ahmed Abbasi and Hsinchun Chen. 2008. Writeprints: A stylometric approach to identity-level identification and similarity detection in cyberspace. *ACM Transactions on Information Systems*, 26(2):7:1–7:29.

Shlomo Argamon, Casey Whitelaw, Paul Chase, Sobhan Raj Hota, Navendu Garg, and Shlomo Levitan. 2007. Stylistic text classification using functional lexical features: Research articles. *J. Am. Soc. Inf. Sci. Technol.*, 58(6):802–822.

Matthew Berland and Eugene Charniak. 1999. Finding parts in very large corpora. In *Proc. of ACL*, pages 57–64, College Park, Maryland, USA.

Ergun Biciçi and Deniz Yuret. 2006. Clustering word pairs to answer analogy questions. In *Proc. of TAINN*, pages 1–8.

Danushka T. Bollegala, Yutaka Matsuo, and Mitsuru Ishizuka. 2009. Measuring the similarity between implicit semantic relations from the web. In *Proc. of WWW*, New York, New York, USA. ACM Press.

Sarah R. Boutwell. 2011. Authorship Attribution of Short Messages Using Multimodal Features. Master's thesis, Naval Postgraduate School.

John Burrows. 2002. 'Delta': a Measure of Stylistic Difference and a Guide to Likely Authorship. *Literary and Linguistic Computing*, 17(3):267–287.

Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Timothy Chklovski and Patrick Pantel. 2004. Verbocean: Mining the web for fine-grained semantic verb relations. In Dekang Lin and Dekai Wu, editors, *Proc. of EMNLP*, pages 33–40, Barcelona, Spain.

Dmitry Davidov and Ari Rappoport. 2006. Efficient unsupervised discovery of word categories using symmetric patterns and high frequency words. In *Proc. of ACL-Coling*, pages 297–304, Sydney, Australia.

Dmitry Davidov and Ari Rappoport. 2008a. Classification of semantic relationships between nominals using pattern clusters. In *Proceedings of ACL-08: HLT*, pages 227–235, Columbus, Ohio, June. Association for Computational Linguistics.

Dmitry Davidov and Ari Rappoport. 2008b. Unsupervised discovery of generic relationships using pattern clusters and its evaluation by automatically generated SAT analogy questions. In *Proc. of ACL-HLT*, pages 692–700, Columbus, Ohio.

Dmitry Davidov, Ari Rappoport, and Moshe Koppel. 2007. Fully unsupervised discovery of concept-specific relationships by web mining. In *Proc. of ACL*, pages 232–239, Prague, Czech Republic.

Dmitry Davidov, Oren Tsur, and Ari Rappoport. 2010a. Semi-supervised recognition of sarcastic sentences in twitter and amazon. In *Proc. of CoNLL*, pages 107–116, Uppsala, Sweden.

Dmitry Davidov, Oren Tsur, and Ari Rappoport. 2010b. Enhanced sentiment learning using twitter hashtags and smileys. In *Proc. of Coling*, pages 241–249, Beijing, China.

Olivier De Vel, Alison Anderson, Malcolm Corney, and George Mohay. 2001. Mining e-mail content for author identification forensics. *ACM Sigmod Record*, 30(4):55–64.

Joachim Diederich, Jörg Kindermann, Edda Leopold, and Gerhard Paass. 2003. Authorship attribution with support vector machines. *Applied intelligence*, 19(1-2):109–123.

Adriano Ferraresi, Eros Zanchetta, Marco Baroni, and Silvia Bernardini. 2008. Introducing and evaluating ukwac, a very large web-derived corpus of english. In *Proc. of the 4th Web as Corpus Workshop*, WAC-4.

Georgia Frantzeskou, Efstathios Stamatatos, Stefanos Gritzalis, and Carole E Chaski. 2007. Identifying authorship by byte-level n-grams: The source code author profile (scap) method. *Int Journal of Digital Evidence*, 6(1):1–18.

Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proc. of Coling – Volume 2*, pages 539–545, Stroudsburg, PA, USA.

Johan F Hoorn, Stefan L Frank, Wojtek Kowalczyk, and Floor van der Ham. 1999. Neural network identification of poets using letter sequences. *Literary and Linguistic Computing*, 14(3):311–338.

Shunichi Ishihara. 2011. A forensic authorship classification in sms messages: A likelihood ratio based approach using n-gram. In *Proc. of the Australasian Language Technology Association Workshop 2011*, pages 47–56, Canberra, Australia.

Patrick Juola. 2012. Large-scale experiments in authorship attribution. *English Studies*, 93(3):275–283.

Bradley Kjell, W Addison Woods, and Ophir Frieder. 1995. Information retrieval using letter tuples with neural network and nearest neighbor classifiers. In *IEEE International Conference on Systems, Man and Cybernetics*, volume 2, pages 1222–1226. IEEE.

Bradley Kjell. 1994. Authorship determination using letter pair frequency features with neural network classifiers. *Literary and Linguistic Computing*, 9(2):119–124.

Moshe Koppel and Jonathan Schler. 2003. Exploiting stylistic idiosyncrasies for authorship attribution. In *Proc. of IJCAI'03 Workshop on Computational Approaches to Style Analysis and Synthesis*, volume 69, page 72.

Moshe Koppel, Jonathan Schler, and Kfir Zigdon. 2005. Determining an author's native language by mining a text for errors. In *Proc. of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, KDD '05, pages 624–628, New York, NY, USA.

Moshe Koppel, Jonathan Schler, Shlomo Argamon, and Eran Messeri. 2006. Authorship attribution with thousands of candidate authors. In *SIGIR*, pages 659–660.

Moshe Koppel, Jonathan Schler, and Elisheva Bonchek-Dokow. 2007. Measuring differentiability: Unmasking pseudonymous authors. *JMLR*, 8:1261–1276.

Moshe Koppel, Jonathan Schler, and Shlomo Argamon. 2009. Computational methods in authorship attribution. *J. Am. Soc. Inf. Sci. Technol.*, 60(1):9–26.

Moshe Koppel, Navot Akiva, Idan Dershowitz, and Nachum Dershowitz. 2011a. Unsupervised decomposition of a document into authorial components. In *Proc. of ACL-HLT*, pages 1356–1364, Portland, Oregon, USA.

Moshe Koppel, Jonathan Schler, and Shlomo Argamon. 2011b. Authorship attribution in the wild. *Language Resources and Evaluation*, 45(1):83–94.

Zornitsa Kozareva, Ellen Riloff, and Eduard Hovy. 2008. Semantic class learning from the web with hyponym pattern linkage graphs. In *Proc. of ACL-HLT*, pages 1048–1056, Columbus, Ohio.

Robert Layton, Paul Watters, and Richard Dazeley. 2010. Authorship attribution for twitter in 140 characters or less. In *Proc. of the 2010 Second Cybercrime and Trustworthy Computing Workshop*, CTC '10, pages 1–8, Washington, DC, USA. IEEE Computer Society.

Robert AJ Matthews and Thomas VN Merriam. 1993. Neural computation in stylometry i: An application to the works of shakespeare and fletcher. *Literary and Linguistic Computing*, 8(4):203–209.

DL Mealand. 1995. Correspondence analysis of luke. *Literary and linguistic computing*, 10(3):171–182.

Thomas Corwin Mendenhall. 1887. The characteristic curves of composition. *Science*, ns-9(214S):237–246.

George K Mikros and Kostas Perifanos. 2013. Authorship attribution in greek tweets using authors multilevel n-gram profiles. In *2013 AAAI Spring Symposium Series*.

Ashwin Mohan, Ibrahim M Baggili, and Marcus K Rogers. 2010. Authorship attribution of sms messages using an n-grams approach. Technical report, CERIAS Tech Report 2011.

Frederick Mosteller and David Lee Wallace. 1964. *Inference and disputed authorship: The Federalist*. Addison-Wesley.

Fuchun Peng, Dale Schuurmans, and Shaojun Wang. 2004. Augmenting naive bayes classifiers with statistical language models. *Information Retrieval*, 7(3-4):317–345.

Conrad Sanderson and Simon Guenter. 2006. Short text authorship attribution via sequence kernels, markov chains and author unmasking: An investigation. In *Proc. of EMNLP*, pages 482–491, Sydney, Australia.

Rui Sousa Silva, Gustavo Laboreiro, Luís Sarmento, Tim Grant, Eugénio Oliveira, and Belinda Maia. 2011. 'twazn me!!! ;(' automatic authorship analysis of micro-blogging messages. In *Proc. of the 16th international conference on Natural language processing and information systems*, NLDB'11, pages 161–168, Berlin, Heidelberg. Springer-Verlag.

Thamar Solorio, Sangita Pillay, Sindhu Raghavan, and Manuel Montes-Gomez. 2011. Modality specific meta features for authorship attribution in web forum posts. In *Proc. of IJCNLP*, pages 156–164, Chiang Mai, Thailand, November.

Efstathios Stamatatos. 2008. Author identification: Using text sampling to handle the class imbalance problem. *Inf. Process. Manage.*, 44(2):790–799.

Efstathios Stamatatos. 2009. A survey of modern authorship attribution methods. *Journal of the American Society for Information Science and Technology*, 60(3):538–556.

Oren Tsur, Dmitry Davidov, and Ari Rappoport. 2010. Icwsm–a great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews. In *Proc. of ICWSM*.

Peter Turney. 2008a. A uniform approach to analogies, synonyms, antonyms, and associations. In *Proc. of Coling*, pages 905–912, Manchester, UK, August. Coling 2008 Organizing Committee.

Peter D. Turney. 2008b. The latent relation mapping engine: Algorithm and experiments. *Journal of Artificial Intelligence Research*, 33:615–655.

Dominic Widdows and Beate Dorow. 2002. A graph model for unsupervised lexical acquisition. In *Proc. of Coling*, pages 1–7, Stroudsburg, PA, USA.

George Udny Yule. 1939. On sentence-length as a statistical characteristic of style in prose: with application to two cases of disputed authorship. *Biometrika*, 30(3-4):363–390.

Chapter 6

Conclusions and Future Work

This dissertation focused on different aspects of lexical semantics. It demonstrated the ability (or lack of ability) of state-of-the-art word embeddings to capture these aspects, and showed that patterns can both serve as better features for lexical semantic tasks, and be integrated into existing word embeddings in order to improve their performance on other tasks.

I started by pointing out to a few limitations of leading word embeddings:

- Their difficulties in distinguishing between *associated* pairs of words (*money*, ***bank***) and *similar* pairs of words (*money*, ***cash***).
- Their inability to distinguish between *similar* pairs of words (*high*, ***tall***) and *opposite* pairs of words (*high*, ***low***).
- Their inferior performance on verb related tasks.

I then showed that lexico-syntactic patterns can serve as useful features for lexical semantic tasks. I presented I-k-NN, a novel, minimally-supervised variant of the k-Nearest-Neighbors algorithm (Chapter 2). I applied it with *symmetric* pattern edge weights to a minimally supervised word classification task. The model obtains an average 86% accuracy results on four classification tasks, while requiring no more than two positive training examples. These results are 16% and 22.5% higher than leading word embeddings. Moreover, the novel I-k-NN algorithm turned out to be more effective than state-of-the-art semi-supervised algorithms, obtaining substantial improvements.

I then showed that patterns can be integrated into state-of-the-art word embeddings, and result in a model that remedies the original models' problems. In Chapter 3, I presented a co-occurrence count model that replaces bag-of-word counts with symmetric pattern counts. The symmetric patterns contexts allowed the model to capture word similarity rather than relatedness. In addition, this chapter introduced a pattern-based feature integrated into the model, which allows it to distinguish between similar and opposite words. In experimenting with the SimLex999 word similarity dataset (Hill et al., 2015), the model obtained 5.5%-16.7% improvement compared to six state-of-the-art models.

Moreover, the model presented in Chapter 3 is able to overcome the third problem of leading word embeddings – their poor performance on verb related tasks. Unlike these models, which suffer a large degradation in performance when shifting from verbs to nouns, the new model performs roughly the same on both word types. This translates to massive 20.2%-40.5% improvements on the verb similarity portion of SimLex999 compared to leading word embeddings.

I then presented (Chapter 4) a novel variant of the word2vec skipgram model (Mikolov et al., 2013b), which replaces bag-of-word contexts with symmetric pattern contexts. The model obtains 15% improvement on verb similarity prediction. Moreover, the model also obtains substantial improvements on adjectives (up to 9%), and is super fast to train, requiring only 2-3% of the training time of the skip-gram model with bag-of-words or dependency contexts.

Finally, for completeness, I presented another application of patterns – an authorship attribution system for very short texts. This pattern-based system obtains state-of-the-art results on the task of identifying the author of Twitter tweets. The system is able to present impressive results (more than 30% accuracy) even in a setup of 1,000 authors. Lastly, the system identified a unique signature for many of the authors, which indicates that authors tend to adopt a unique style even when writing very short texts.

In addition to the chapters of this dissertation, I also took part in a few related projects. In (Rubinstein et al., 2015), we pointed out to another limitation of leading word embeddings – their inability to capture *attributive* properties (e.g., bananas are *yellow*, elephants are *big*). We selected several attributive properties (e.g., *is_red*) and showed that a classifier that uses state-of-the-art embeddings as features is unable to separate between words that have this property (e.g., *strawberry*) and those that do not have it (e.g., *table*). The poor classification performance is contrasted against a relatively high performance on learning *taxonomic* properties (e.g., *dog* is an *animal*, *apple* is a *fruit*).

In (Vulić et al., 2017), we extended the work described in Chapter 4, and developed an automatic method for extracting the most suitable context types for each word category. Our results show that our method can improve word similarity prediction scores on verbs, which are, as shown in this dissertation, a known caveat for word embeddings, but also for nouns and adjectives, on which leading models perform relatively well.

To conclude, this dissertation demonstrated that patterns are a very effective tool for representing lexical semantics, one that is able to overcome many of the problems that word embeddings that rely on bag-of-word contexts suffer from. I have shown the power of patterns both in the context of existing models, and in the context of novel algorithms. The pattern-based models obtain state-of-the-art results in all cases.

Future Work

The work presented in this dissertation can serve as basis for many future directions. The first direction is finding better ways to exploit the information captured by patterns. Bag-of-words, dependency edges and patterns capture different types of information. In Chapter 3 I have presented initial results that indicate that their strengths are complementary. In (Vulić et al., 2017) we presented a method for combining different context types in order to improve the representation of different part-of-speech types. A question that remains open is how to construct a principled model for integrating patterns and other types of contexts. A related research trend studied the integration into beg-of-words word embeddings of external knowledge from manually constructed resources such as WordNet (Kiela et al., 2015; Pham et al., 2015; Liu et al., 2015; Faruqui et al., 2015; Mrkšić et al., 2016). Future work will include designing a specific model for making the most of the combination between patterns and other context types.

Another direction concerns multilingualism. Although this dissertation focuses on English, the strength of patterns also comes from their corpus based nature, which makes them applicable to many other languages (Davidov and Rappoport, 2006, 2010). The prevalence of evaluation datasets in other languages (Hassan and Mihalcea, 2009; Joubarne and Inkpen, 2011; Leviant and Reichart, 2015) calls for testing whether the problems reported in this dissertation, as well as their pattern-based solutions, generalize cross-linguistically.

An immediate, highly studied direction is looking into other lexical semantic tasks. Patterns have been shown useful for capturing a range of such tasks (Hearst, 1992; Lin et al., 2003; Snow et al., 2004; Davidov and Rappoport, 2008a), while in contrast, embeddings have not been shown to perform as well (Levy et al., 2015a). The question that remains open is can these methods be combined to improve performance on tasks such as hypernymy or entailment detection.

Another issue that arises from this work is its applicability to downstream applications. One of the main reasons for doing research on word embeddings is that embeddings have been shown useful to a range of NLP tasks, most notably as vector initialization for NN models (Socher et al., 2013; Chen and Manning, 2014; Devlin et al., 2014). Several works have shown that high evaluation scores on intrinsic dataset do not directly translate to improved performance on extrinsic tasks (Schnabel et al., 2015; Tsvetkov et al., 2015; Melamud et al., 2016). A promising future direction is to check whether the improved word similarity performance (and verb similarity in particular) presented here could translate to improved performance for tasks like parsing, machine translation and sentiment analysis.

An important question that remains open is whether "the perfect embeddings" even exist. In the beginning of this dissertation I provided the following quote: "*all of the semantics of human language might one day be captured in some kind of Vector Space Model*" (Turney and Pantel, 2010). The work presented here questions this hypothesis. Some of the issues raised here are not necessarily problems, and one could imagine a downstream application that would actually find them useful. For example, distinguishing between similar and related concepts is irrelevant for document classification systems. Similarly, word classification systems might consider antonyms in the same semantic category (e.g., "tall" and "short" are both size adjectives). As the cosine similarity between "cup" and "coffee" (or between "good" and "bad") cannot be both high and low, each model must take a pick, and cannot support both alternatives. This means that at the very least, the current method for applying word embeddings – reducing them to a single number (cosine similarity) – cannot suffice for a complete representation of lexical semantics. An immediate corollary is that research should focus not only on building stronger and more sophisticated models, but also on learning how to better exploit the information encompassed in them.

Finally, this dissertation focused on *patterns* as a useful tool for representing words, as well as other tasks such as authorship attribution. The term "pattern" has been used to term various different linguistic objects. In this work I used at least two different types of patterns (symmetric pattern, and general high frequency/low frequency patterns). Other types exist, including dependency patterns (Baroni and Lenci, 2010), word patterns (Turney, 2006, 2008b) and PoS patterns (Allan and Raghavan, 2002). Future work will compare the different types of patterns and establish which type (if any) is preferable for word representation tasks, as well as other tasks for which they are useful, such as sentiment analysis and authorship attribution.

Bibliography

Abbasi, A. and Chen, H. (2005). Applying authorship analysis to extremist-group web forum messages. *IEEE Intelligent Systems*, 20:67–75.

Abbasi, A. and Chen, H. (2008). Writeprints: A stylometric approach to identity-level identification and similarity detection in cyberspace. *ACM Transactions on Information Systems*, 26(2):7:1–7:29.

Abend, O., Reichart, R., and Rappoport, A. (2010). Improved unsupervised pos induction through prototype discovery. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1298–1307. Association for Computational Linguistics.

Allan, J. and Raghavan, H. (2002). Using part-of-speech patterns to reduce query ambiguity. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 307–314. ACM.

Almuhareb, A. (2006). *Attributes in Lexical Acquisition.* PhD thesis, University of Essex.

Argamon, S., Whitelaw, C., Chase, P., Hota, S. R., Garg, N., and Levitan, S. (2007). Stylistic text classification using functional lexical features: Research articles. *J. Am. Soc. Inf. Sci. Technol.*, 58(6):802–822.

Baroni, M., Dinu, G., and Kruszewski, G. (2014). Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proc. of ACL.*

Baroni, M., Evert, S., and Lenci, A. (2008). Bridging the gap between semantic theory and computational simulations: Proceedings of the esslli workshop on distributional lexical semantics. *Language and Information (FoLLI).*

Baroni, M. and Lenci, A. (2010). Distributional memory: A general framework for corpus-based semantics. *Computational Linguistics*, 36(4):673–721.

Baroni, M., Murphy, B., Barbu, E., and Poesio, M. (2010). Strudel: A corpus-based semantic model based on properties and types. *Cognitive Science*, 34(2):222–254.

Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. (2003). A neural probabilistic language model. *JMLR*.

Berland, M. and Charniak, E. (1999). Finding parts in very large corpora. In *Proc. of ACL*.

Bollegala, D., Maehara, T., Yoshida, Y., and ichi Kawarabayashi, K. (2015). Learning word representations from relational graphs. In *Proc. of AAAI*.

Boutwell, S. R. (2011). Authorship Attribution of Short Messages Using Multimodal Features. Master's thesis, Naval Postgraduate School.

Bruni, E., Tran, N.-K., and Baroni, M. (2014). Multimodal distributional semantics. *JAIR*.

Chen, D. and Manning, C. D. (2014). A fast and accurate dependency parser using neural networks. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Christodoulopoulos, C., Goldwater, S., and Steedman, M. (2010). Two decades of unsupervised pos induction: How far have we come? In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Clark, S. (2012). Vector space models of lexical meaning. *Handbook of Contemporary Semanticssecond edition*, pages 1–42.

Collins, M. (2003). Head-driven statistical models for natural language parsing. *Computational linguistics*, 29(4):589–637.

Collobert, R. and Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proc. of ICML*.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *JMLR*, 12:2493–2537.

Davidov, D. and Rappoport, A. (2006). Efficient unsupervised discovery of word categories using symmetric patterns and high frequency words. In *Proc. of ACL-COLING*.

Davidov, D. and Rappoport, A. (2008a). Classification of semantic relationships between nominals using pattern clusters. In *Proceedings of ACL-08: HLT*, pages 227–235, Columbus, Ohio. Association for Computational Linguistics.

Davidov, D. and Rappoport, A. (2008b). Unsupervised discovery of generic relationships using pattern clusters and its evaluation by automatically generated SAT analogy questions. In *Proc. of ACL-HLT*.

Davidov, D. and Rappoport, A. (2010). Automated translation of semantic relationships. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 241–249, Beijing, China. Coling 2010 Organizing Committee.

Davidov, D., Tsur, O., and Rappoport, A. (2010). Enhanced sentiment learning using twitter hashtags and smileys. In *Proc. of COLING*.

De Vel, O., Anderson, A., Corney, M., and Mohay, G. (2001). Mining e-mail content for author identification forensics. *ACM Sigmod Record*, 30(4):55–64.

Devlin, J., Zbib, R., Huang, Z., Lamar, T., Schwartz, R. M., and Makhoul, J. (2014). Fast and robust neural network joint models for statistical machine translation. In *ACL (1)*, pages 1370–1380. Citeseer.

Dhillon, P., Foster, D. P., and Ungar, L. H. (2011). Multi-view learning of word embeddings via cca. In *Proc. of NIPS*.

Erk, K. (2012). Vector Space Models of Word Meaning and Phrase Meaning: A Survey. *Language and Linguistics Compass*, 6(10):635–653.

Etzioni, O., Cafarella, M., Downey, D., Popescu, A.-M., Shaked, T., Soderland, S., Weld, D. S., and Yates, A. (2005). Unsupervised named-entity extraction from the web: An experimental study. *Artificial intelligence*, 165(1):91–134.

Faruqui, M., Dodge, J., Jauhar, S. K., Dyer, C., Hovy, E., and Smith, N. A. (2015). Retrofitting word vectors to semantic lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1606–1615, Denver, Colorado. Association for Computational Linguistics.

Faruqui, M., Tsvetkov, Y., Rastogi, P., and Dyer, C. (2016). Problems with evaluation of word embeddings using word similarity tasks. In *Proc. of the 1st Workshop on Evaluating Vector Space Representations for NLP.*

Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., and Ruppin, E. (2001). Placing search in context: The concept revisited. In *Proc. of WWW*, pages 406–414. ACM.

Frantzeskou, G., Stamatatos, E., Gritzalis, S., and Chaski, C. E. (2007). Identifying authorship by byte-level n-grams: The source code author profile (scap) method. *Int Journal of Digital Evidence*, 6(1):1–18.

Freitag, D., Blume, M., Byrnes, J., Chow, E., Kapadia, S., Rohwer, R., and Wang, Z. (2005). New experiments in distributional representations of synonymy. In *In Proc. of CoNLL.*

Grishman, R., Macleod, C., and Meyers, A. (1994). Comlex syntax: Building a computational lexicon. In *Proceedings of the 15th conference on Computational linguistics-Volume 1*, pages 268–272. Association for Computational Linguistics.

Harris, Z. (1954). Distributional structure. *Word.*

Hassan, S. and Mihalcea, R. (2009). Cross-lingual semantic relatedness using encyclopedic knowledge. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*, pages 1192–1201. Association for Computational Linguistics.

Hearst, M. A. (1992). Automatic acquisition of hyponyms from large text corpora. In *Proc. of COLING – Volume 2.*

Hill, F., Reichart, R., and Korhonen, A. (2015). Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695.

Ishihara, S. (2011). A forensic authorship classification in sms messages: A likelihood ratio based approach using n-gram. In *Proc. of the Australasian Language Technology Association Workshop 2011.*

Joubarne, C. and Inkpen, D. (2011). Comparison of semantic similarity for different languages using the google n-gram corpus and second-order cooccurrence measures. In *Canadian Conference on Artificial Intelligence*, pages 216–221. Springer.

Kiela, D., Hill, F., and Clark, S. (2015). Specializing word embeddings for similarity or relatedness. In *Proc. of EMNLP*.

Klein, D. and Manning, C. D. (2004). Corpus-based induction of syntactic structure: Models of dependency and constituency. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 478. Association for Computational Linguistics.

Koehn, P. (2009). *Statistical machine translation*. Cambridge University Press.

Koppel, M., Akiva, N., Dershowitz, I., and Dershowitz, N. (2011a). Unsupervised decomposition of a document into authorial components. In *Proc. of ACL-HLT*.

Koppel, M. and Schler, J. (2003). Exploiting stylistic idiosyncrasies for authorship attribution. In *Proc. of IJCAI'03 Workshop on Computational Approaches to Style Analysis and Synthesis*.

Koppel, M., Schler, J., and Argamon, S. (2011b). Authorship attribution in the wild. *Language Resources and Evaluation*, 45(1):83–94.

Koppel, M., Schler, J., Argamon, S., and Messeri, E. (2006). Authorship attribution with thousands of candidate authors. In *SIGIR*.

Koppel, M., Schler, J., and Bonchek-Dokow, E. (2007). Measuring differentiability: Unmasking pseudonymous authors. *JMLR*, 8:1261–1276.

Kübler, S., McDonald, R., and Nivre, J. (2009). *Dependency Parsing*. Morgan And Claypool Publishers.

Lebret, R. and Collobert, R. (2014). Word embeddings through hellinger pca. In *Proc. of EACL*.

Leviant, I. and Reichart, R. (2015). Separated by an un-common language: Towards judgment language informed vector space modeling.

Levy, O. and Goldberg, Y. (2014). Dependency-based word embeddings. In *Proc. of ACL*.

Levy, O., Goldberg, Y., and Dagan, I. (2015a). Improving distributional similarity with lessons learned from word embeddings. *TACL*.

Levy, O., Remus, S., Biemann, C., and Dagan, I. (2015b). Do supervised distributional methods really learn lexical inference relations? In *Proc. of NAACL*.

Lin, D. (1998). Automatic retrieval and clustering of similar words. In *Proc. of ACL*.

Lin, D., Zhao, S., Qin, L., and Zhou, M. (2003). Identifying synonyms among distributionally similar words. In *Proc. of IJCAI*.

Liu, Q., Jiang, H., Wei, S., Ling, Z.-H., and Hu, Y. (2015). Learning semantic word embeddings based on ordinal knowledge constraints. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1501–1511, Beijing, China. Association for Computational Linguistics.

Mani, I. and Maybury, M. T. (1999). *Advances in automatic text summarization*, volume 293. MIT Press.

McDonald, R., Pereira, F., Ribarov, K., and Hajič, J. (2005). Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 523–530. Association for Computational Linguistics.

Mealand, D. (1995). Correspondence analysis of luke. *Literary and linguistic computing*, 10(3):171–182.

Melamud, O., McClosky, D., Patwardhan, S., and Bansal, M. (2016). The role of context types and dimensionality in learning word embeddings. In *Proc. of NAACL*.

Mendenhall, T. C. (1887). The characteristic curves of composition. *Science*, ns-9(214S):237–246.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Proc. of NIPS*.

Mikolov, T., Yih, W.-t., and Zweig, G. (2013c). Linguistic regularities in continuous space word representations. In *Proc. of NAACL-HLT*.

Mikros, G. K. and Perifanos, K. (2013). Authorship attribution in greek tweets using authors multilevel n-gram profiles. In *2013 AAAI Spring Symposium Series.*

Miller, G. A. (1995). Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Miller, G. A. and Charles, W. G. (1991). Contextual correlates of semantic similarity. *Language and cognitive processes.*

Mnih, A. and Hinton, G. E. (2009). A scalable hierarchical distributed language model. In *Proc. of NIPS.*

Mnih, A. and Kavukcuoglu, K. (2013). Learning word embeddings efficiently with noise-contrastive estimation. In *Proc. of NIPS.*

Mohan, A., Baggili, I. M., and Rogers, M. K. (2010). Authorship attribution of sms messages using an n-grams approach. Technical report, CERIAS Tech Report 2011.

Mosteller, F. and Wallace, D. L. (1964). *Inference and disputed authorship: The Federalist.* Addison-Wesley.

Mrkšić, N., Ó Séaghdha, D., Thomson, Milica Gašić, B., Rojas-Barahona, L., Su, P.-H., Vandyke, D., Wen, T.-H., and Young, S. (2016). Counter-fitting word vectors to linguistic constraints. In *Proc. of NAACL.*

Murphy, B., Talukdar, P. P., and Mitchell, T. (2012). Learning Effective and Interpretable Semantic Models using Non-Negative Sparse Embedding. In *Proc. of COLING.*

Navigli, R. and Ponzetto, S. P. (2012). Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250.

Nivre, J. and Hall, J. (2005). MaltParser: A language-independent system for data-driven dependency parsing. In *Proceedings of the Fourth Workshop on Treebanks and Linguistic Theories (TLT)*, pages 137–148.

Padó, S. and Lapata, M. (2007). Dependency-based construction of semantic space models. *Computational Linguistics.*

Palmer, M., Gildea, D., and Xue, N. (2010). *Semantic role labeling*, volume 3. Morgan & Claypool Publishers.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proc. of EMNLP.*

Pham, N. T., Lazaridou, A., and Baroni, M. (2015). A multitask objective to inject lexical contrast into distributional semantics. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 21–26, Beijing, China. Association for Computational Linguistics.

Rubenstein, H. and Goodenough, J. B. (1965). Contextual correlates of synonymy. *Communications of the ACM.*

Rubinstein, D., Levi, E., Schwartz, R., and Rappoport, A. (2015). How well do distributional models capture different types of semantic knowledge? In *Proc. of ACL.*

Salton, G. (1971). *The SMART Retrieval System: Experiments in Automatic Document Processing.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Schnabel, T., Labutov, I., Mimno, D., and Joachims, T. (2015). Evaluation methods for unsupervised word embeddings. In *In Proc. of EMNLP.*

Schuler, K. K. (2005). *VerbNet: A broad-coverage, comprehensive verb lexicon.* PhD thesis, University of Pennsylvania.

Schwartz, R., Reichart, R., and Rappoport, A. (2014). Minimally supervised classification to semantic categories using automatically acquired symmetric patterns. In *Proc. of COLING.*

Schwartz, R., Tsur, O., Rappoport, A., and Koppel, M. (2013). Authorship attribution of micro-messages. In *Proc. of EMNLP.*

Seginer, Y. (2007). Fast unsupervised incremental parsing. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 384–391, Prague, Czech Republic. Association for Computational Linguistics.

Silva, R. S., Laboreiro, G., Sarmento, L., Grant, T., Oliveira, E., and Maia, B. (2011). 'twazn me!!! ;(' automatic authorship analysis of micro-blogging messages. In *Proc. of NLDB.*

Snow, R., Jurafsky, D., and Ng, A. Y. (2004). Learning syntactic patterns for automatic hypernym discovery. *Advances in Neural Information Processing Systems 17.*

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., and Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proc. of EMNLP*.

Solorio, T., Pillay, S., Raghavan, S., and Montes-Gomez, M. (2011). Modality specific meta features for authorship attribution in web forum posts. In *Proc. of IJCNLP*.

Suchanek, F. M., Kasneci, G., and Weikum, G. (2007). Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706. ACM.

Toutanova, K., Klein, D., Manning, C. D., and Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proc. of NAACL*.

Tsur, O., Davidov, D., and Rappoport, A. (2010). Icwsm–a great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews. In *Proc. of ICWSM*.

Tsvetkov, Y., Faruqui, M., Ling, W., Lample, G., and Dyer, C. (2015). Evaluation of word vector representations by subspace. In *In Proc. of EMNLP*.

Turian, J., Ratinov, L., and Bengio, Y. (2010). Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 384–394. Association for Computational Linguistics.

Turney, P. D. (2006). Similarity of semantic relations. *Computational Linguistics*.

Turney, P. D. (2008a). A uniform approach to analogies, synonyms, antonyms, and associations. In *Proc. of COLING*.

Turney, P. D. (2008b). A uniform approach to analogies, synonyms, antonyms, and associations. In *Proc. of COLING*.

Turney, P. D. and Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence research*, 37(1):141–188.

Vossen, P. (1998). *A multilingual database with lexical semantic networks*. Springer.

Vulić, I., Schwartz, R., Rappoport, A., Reichart, R., and Korhonen, A. (2017). Automatic selection of context configurations for improved (and fast) class-specific word representations. In *Proc. of CoNLL*.

Widdows, D. and Dorow, B. (2002). A graph model for unsupervised lexical acquisition. In *Proc. of COLING*.

Yatbaz, M. A., Sert, E., and Yuret, D. (2012). Learning syntactic categories using paradigmatic representations of word context. In *EMNLP-CoNLL*, pages 940–951.

Yule, G. U. (1939). On sentence-length as a statistical characteristic of style in prose: with application to two cases of disputed authorship. *Biometrika*, 30(3-4):363–390.

Zelle, J. M. and Mooney, R. J. (1996). Learning to parse database queries using inductive logic programming. In *Proceedings of the national conference on artificial intelligence*, pages 1050–1055.

Zettlemoyer, L. S. and Collins, M. (2005). Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *UAI*, pages 658–666. AUAI Press.

תקציר

עיבוד שפה (Natural Language Processing) הוא תחום מחקר שמנסה, מחד, לתת פתרונות חישוביים לשאלות מחקר בלשניות, ומאידך, לפתח יישומים למשימות הקשורות לשפה, כגון תרגום מכונה, תמצות אוטומטי ומענה על שאלות. שתי המטרות הללו חולקות שאלה מרכזית משותפת, והיא כיצד לייצג משמעות (סמנטיקה); למשל, מה המשמעות של המילה *כלב*, של ביטויים כמו *שטיח אדום* או *לאכול את הכובע*, או של מבני שפה מורכבים יותר. שאלה זו היא במהותה בלשנית (או אפילו קוגניטיבית), ועם זאת יש לה השלכות פרקטיות ואמפיריות.

השיטה הנפוצה ביותר לייצוג המשמעות של אלמנטים של שפה היא ע״י בניית וקטורים של תכוניות (feature vectors). שיטות אלו, הנקראות גם *vector space models*, פותחו לראשונה כבר בראשית שנות ה-70. עד לאחרונה, מודלים מסוג vector space models בנו מטריצת שכנויות, כך שכל מילה מיוצגת על ידי המילים האחרות שאיתן היא נוטה להופיע בטקסט. בשנים האחרונות פותחו שיטות חדשניות לבניית וקטורי תכוניות. שיטות אלו נקראות שיכוני מילים (*word embeddings*). מודלים אלו, המבוססים לרוב על אלגוריתמים מבוססי רשתות נוירונים, הובילו לשיפורים משמעותיים במגוון משימות סמנטיות. הצלחה זו הפכה את שיכוני המילים לכלי מאוד פופולרי, ויצרה תחושה שיש ביכולתם לייצג את השדה הסמנטי המלא של כל מילה בלקסיקון.

בעבודה זו, אני אראה שלמרות הצלחתם הכבירה, שיכוני מילים סובלים ממספר מוגבלויות. ראשית, אני אראה שלמרות ששיכוני המילים המובילים בימינו מצליחים לתפוס בצורה יוצאת מן הכלל קשר של *אסוציאציה* בין מילים (הקשר בין *פרה* ל-***חלב***), הם הרבה פחות טובים בזיהוי *דמיון* בין מילים (*פרה* / ***סוס***). שנית, אני אדגים שהם לא מסוגלים להבחין בין מילים *דומות* (*טוב* / ***מעולה***) לבין מילים *הפוכות* (*טוב* / ***רע***). שלישית, אני אראה שבעוד ששיכוני מילים הינם כלי מאוד מוצלח לייצוג המשמעות של *שמות עצם* (למשל, *כלב* או *בית*), הם הרבה פחות מוצלחים בייצוג *פעלים* (*לאכול*, *לרוץ*).

על מנת להתמודד עם הבעיות הללו, אני אציג מספר פתרונות מבוססי תבניות (*פטרנים*, לדוגמא "X such as Y", "X is a Y"). תבניות הן אחת האלטרנטיבות הכי מוצלחות לשיכוני מילים. הן הוכחו כמוצלחות בזיהוי מגוון רחב של יחסים סמנטיים. אני אראה ששילובן בתוך תבניות שיכון יכול להקל במידה רבה על הבעיות של האחרונים.

אני אתחיל בלהדגים שתבניות יכולות לשמש כתכוניות יותר מוצלחות מאשר שיכוני מילים עבור משימות סמנטיות. אני אציג הרחבה של אלגוריתם ה-k-Nearest Neighbors, ומודל שמשתמש בהרחבה זו עם תכוניות מבוססות תבניות סימטריות (למשל, "X and Y", "X or Y") על מנת לזהות מגוון תכונות סמנטיות (למשל, האם ניתן לאכול את האובייקט, האם האובייקט חי או לא), ומגיע לשיפורים משמעותיים יחסית לשיכוני מילים מובילים.

אני אמשיך בהצגת שני מודלים של שיכוני מילים אשר מבוססים על תבניות סימטריות, ואשר מסוגלים להתגבר על הבעיות של שיכוני מילים. הראשון הוא מודל המגיע לתוצאות הגבוהות בצורה משמעותית מהתוצאות של שישה מודלים מובילים במשימת חיזוי דמיון בין מילים. השני הוא גירסא מבוססת תבניות של המודל המפורסם word2vec skipgram (Mikolov et al., 2013b), אשר משיגה תוצאות הגבוהות ב-15% מהמודל המקורי במשימת חיזוי דמיון בין פעלים, זאת למרות שזמן האימון שלה קצר בצורה דרמטית יחסית לזמן האימון של המודל המקורי.

לבסוף, אציג עבודה נוספת מבוססת תבניות, שמראה שהן יכולות לשמש כתכוניות בעלות ערך גם למשימה נוספת – זיהוי הכותב של ציוץ (tweet) בודד.

לסיכום, התרומה של עבודה זו באה לידי ביטוי בשני אספקטים. ראשית, היא שופכת אור על המגבלות (כמו גם על החוזקות) של שיכוני מילים מובילים, אשר נחשבו עד לאחרונה כל-יכולים. שנית, היא מדגימה כיצד ניתן להשתמש בתבניות על מנת להתגבר על מגבלות אלו, הן על ידי

שילוב תכוניות מבוססות תבניות בתוך מודלים קיימים והן על ידי פיתוח מודלים חדשניים, מבוססי תבניות.

עבודה זו נעשתה תחת הדרכתו של

פרופ' ארי רפופורט

שיטות מבוססות תבניות לשיפור סמנטיקה לקסיקלית ושיכוני מילים

חיבור לשם קבלת תואר "דוקטור לפילוסופיה"

מאת: רועי שוורץ

מוגש לסנאט האוניברסיטה העברית בירושלים

יולי, 2016