

Analysis of Context Embeddings in Word Sense Induction

by

Osman Başkaya

A Thesis Submitted to the

Graduate School of Sciences and Engineering

in Partial Fulfillment of the Requirements for

the Degree of

Master of Science

in

Computer Sciences and Engineering

Koç University

September 2015

Acknowledgement

I would like to thank my supervisor Deniz Yuret for his guidance and support. I thank Arzucan Özgür and Engin Erzin for participating in committee. Thanks to Sarah J. Bell for proofreading the thesis. I would like to acknowledge graduate fellowship provided by Scientific and Technological Research Council of Turkey (TUBITAK).

Abstract

There exist several drawbacks of representing the word senses with a fixed list of definitions of a manually constructed lexical database. There is no guarantee that they reflect the exact meaning of a target word in a given context, since they usually contain definitions that are too general. More so, lexical databases often include many rare senses while missing corpus/domain-specific senses. Word Sense Induction (WSI) focuses on discriminating the usages of a polysemous word without using a fixed list of definitions or any hand-crafted resources.

In contrast to the most common approach in WSI, which is to apply clustering or graph partitioning on a representation of first- or second-order co-occurrences of a word, my method obtains a probability distribution for each context suggested by a statistical model. This distribution helps to create context embeddings using the co-occurrence framework that represents the context with low-dimensional, dense vectors in Euclidean space. Then, these context embeddings are clustered by k -means clustering algorithm to discriminate usages (senses) of a word. This method proved its usefulness in Unsupervised Part-of-Speech Induction, and supervised tasks such as Multilingual Dependency Parsing. I examine this method on SemEval 2010 and SemEval 2013 Word Sense Induction lexical sample tasks, and the dataset I created using OntoNotes 5.0. This new lexical sample dataset has high inter-annotator agreement (IAA) ($>90\%$) and number of instances for each word type is more than any previous lexical sample tasks (>500 instances).

The contributions in this thesis are as follows: (1) I suggest a method to attack

the Word Sense Induction problem. (2) I provide a comprehensive analysis (a) in embedding step by comparing other popular word embeddings by transforming each of them to context embeddings using substitute word distributions for each context, and (b) in clustering step by comparing different clustering algorithms (k -means, Spectral Clustering, DBSCAN) and different clustering approaches (*local approach* where instances of each word type clustered separately, and *part-of-speech based approach* where instances tagged with same-part-of-speech clusters independently).

The code to replicate the results in this thesis can be found at <https://github.com/osmanbaskaya/wsid>.

Özetçe

Bir kelimenin anlamlarını elle hazırlanmış sözcüksel bir veritabanı ile temsil etmenin birden çok problemi bulunmaktadır. Bu tip sözlükler bir kelimeye ait tahmin ettiğimiz anlamı işaret edecek bir kayıt bulundurmayaabilirler; zirâ sözlükler çok genel hazırlanmaktadır. Bu sözlükler bir kelimenin çok nâdir kullanılan anlamlarını içerirler fakat alana özgü anlamları barındırmayabilirler. Sözcük anlamı tümevarımı (word sense induction) kelimenin anlamlarını (kullanımlarını) ayırırken elle oluşturulmuş, durağan bir kaynak yardımı almadığından bu tip problemleri çözmektedir.

Diğer popüler çözümlerin (muğlaklığı giderilmek istenen kelimenin birinci dereceden ya da ikinci dereceden birlikte görüldüğü diğer kelimeleri kullanarak elde edilen bir temsil üzerine kümeleme ya da çizge bölümlemesi gibi) aksine tezde takip ettiğimiz yöntem; muğlaklığını gidermek istediğimiz kelimeyle aynı bağlamda kullanılması ihtimâli yüksek olan kelimeleri ve bu kelimelerin olasılıklarını bir dil modeli yardımıyla bulup, bunları S-CODE adlı Öklitsel gömme (embedding) algoritması vasıtasıyla modellemektir. Bu algoritma ile kelime türü gömülerine (word embedding) ek olarak, kullandığımız yöntemle bağlam gömülerini (context embedding) de elde etmekteyiz. Bağlam gömüleri daha sonra k -means algoritması yardımıyla kümelenererek hedef kelimemize ait farklı kullanımlar ayrıştırılmaktadır. Bu yöntemin benzerleri önceki yıllarda gözetimsiz sözcük türü tümevarımı (unsupervised part-of-speech induction) ve gözetimli bağıllık ayrıştırması problemlerinde başarılı olmuşlardır. Yöntemimizi SemEval 2010, SemEval 2013 kelime anlamı tümevarımı yarışmalarında hazırlanmış veri kümelerinde ve OntoNotes projesinin

(versiyon 5.0) işaretlediği cümleleri kullanarak bizim oluşturduğumuz, yüksek uzlaşmaya ($>90\%$) ve en az 500 örneğe sahip kelimelerin bulunduğu bir sözcük veri kümesinde test ettik.

Bu tezin literatüre katkısı şöyle özetlenebilir. (1) Sözcük anlamı tümevarımı problemi için bağlam temsillerinden yararlanan bir yöntem öneriyoruz. (2) (a) Literatürde önerilen kelime temsillerini, aynı bağlamda kullanılması ihtimâli yüksek olan kelimelerin olasılıklarını kullanarak bağlam temsillerine dönüştürüp kendi önerdiğimiz bağlam temsilleriyle kıyasladık. (b) Farklı kümeleme algoritmalarını (k -means, Spectral Clustering, DBSCAN) ve bu algoritmaların Sözcük Anlamı Tümevarımı problemine özel farklı yaklaşımlarını (lokâl yaklaşım, sözcük türüne bağlı yaklaşım) kıyasladık. Son olarak, OntoNotes projesini kullanarak oluşturduğumuz veri kümesini hazırlayan prosedürü, ilerde aynı problemle uğraşacak araştırmacılara yardımı dokunması ümidiyle paylaştık. Araştırmacılar aynı parametrelerle aynı veri kümesini üretip, bizim önerdiğimiz yöntem ile kendi sistemlerini karşılaştırabilir; yâhut farklı parametrelerle kendilerine uyacak bir veri kümesi üretebilirler. Yapılan çalışmayı ve alınan sonuçları tekrarlamak için gerekli kodlara <https://github.com/osmanbaskaya/wsid> adresinden ulaşılabilir.

Contents

Acknowledgement	i
Abstract	ii
Özetçe	iv
List Of Tables	ix
List Of Figures	x
Glossary	xi
1 Introduction	1
1.1 Word Sense Induction and Disambiguation	1
1.2 Thesis	3
1.3 Motivation	3
1.4 Outline	4
2 Related Work	6
2.1 Representations	6
2.2 Techniques	9
2.2.1 Clustering Models	9
2.2.2 Graph-based Models	10
2.2.3 Bayesian Approaches	11
2.2.4 Frequent-termsets Models	12

3	Evaluation Framework	13
3.1	Unsupervised Evaluation	14
3.1.1	FScore	14
3.1.2	Paired FScore	15
3.1.3	V-Measure	17
3.1.4	Fuzzy B-Cubed	18
3.1.5	Fuzzy Normalized Mutual Information	20
3.2	Supervised Evaluation	20
3.2.1	Agirre’s method	20
4	Method	21
4.1	Data Enrichment	21
4.2	Substitute Word Distributions	22
4.3	Discretization of the Substitute Word Distribution	24
4.4	Co-occurrence Modeling with S-CODE algorithm	26
4.5	Clustering	29
5	Experiments	30
5.1	Baselines	30
5.2	Different Representations	32
5.2.1	From Word Representation to Context Representation	32
5.2.2	Comparison of Different Context Representations	33
5.3	Variations in the Clustering step	35
5.3.1	Different Clustering Approaches	36
5.3.2	Different Clustering Algorithms	37
5.4	Comparison between Our System and Other Systems	38
5.4.1	SemEval 2010 WSI task	39
5.4.2	SemEval 2013 WSI task	40
6	Conclusion	46
	Appendices	48

A Datasets	49
A.1 SemEval 2010 WSI Task	49
A.2 SemEval 2013 WSI Task	50
A.3 OntoNotes Sense Dataset	52

List of Tables

3.1	Contingency table	16
4.1	Top 10 substitutes and their probabilities for a context	24
4.2	An example output of discretization step	25
5.1	Random baselines	31
5.2	Comparisons of other embeddings on SemEval 2010 test data	34
5.3	Comparison of other embeddings on SemEval 2013 test data	35
5.4	Performances of Different Clustering Approaches	36
5.5	Performances of Different Clustering Algorithms in SemEval 2010 and SemEval 2013	37
5.6	Comparison between our system and other systems on SemEval 2010	38
5.7	Comparison between our system and other systems on SemEval 2013	43
A.1	Dataset statistics for the SemEval 2010 WSID task	49
A.2	Dataset statistics for the SemEval 2013 WSID task	51
A.3	Dataset statistics for OntoNotes Dataset	53

List of Figures

2.1	Syntactic and paradigmatic axes	7
3.1	Precision and recall calculation for B-Cubed	19
4.1	Illustration of S-CODE algorithm	26
5.1	Context embeddings with different k values (Agirre’s method) . . .	39
5.2	Context embeddings over different cluster sizes (k) (FCubed and FNMI)	40
5.3	Context embeddings with different k values (Agirre’s method) . . .	44
5.4	Context embeddings over different cluster sizes (k) (FCubed and FNMI)	45
A.1	OntoNotes Sense Annotation Procedure	53

Glossary

context is the ordered sequence of word tokens that surrounds a target word token. Context can be a window with a specific size, a sentence, a document.

context representation is the representation of the context of a word token. If the contexts of two different word tokens are the same (same sequence of word tokens), they would have the same context representations. Context representations are independent of the target word the contexts surround.

gold standard is the list of answers for a dataset annotated by humans, used for performance evaluation on a particular task. The type of the answers depends on the task. In Word Sense Induction, gold standard consists of correct senses, one for each instance.

representation is a mathematical object associated with each word, often a vector. A vector representation can be sparse and high-dimensional or dense and low-dimensional. Our word representations are a dense real vector in the d dimensional Euclidean space that is associated with word types.

target word (or target word token) is the word token of interest to identify or distinguish its meaning for a given context.

word token (or token) is an instance (observation) of a word type.

word type (or type) is a unique word (sequence of characters including numbers and punctuation) in a vocabulary.

Chapter 1

Introduction

In this chapter, I will introduce Word Sense Disambiguation and Word Sense Induction problems and summarize the work in the thesis. I will continue by describing the motivations. At the end of the chapter I will give an outline for the rest of the thesis.

1.1 Word Sense Induction and Disambiguation

Word Sense Disambiguation (WSD) is the task of identifying one of the correct senses in a fixed list of definitions (e.g. dictionary) of a polysemous word for a given context. The difficulty of this problem is based on ambiguity in the language. It is not a big surprise that this process is non-trivial for computers, since it can sometimes be hard for a human who has a very accurate and up-to-date built-in language model and enormous knowledge regarding the real world to distinguish similar senses of a word. Thus, some researchers argued that WSD is AI-complete, meaning that the difficulty in solving this problem is equivalently hard to solving the most difficult problems in Artificial Intelligence ([Mallery \(1988\)](#)). Here is an example of ambiguity in natural language:

1. Oğuz bought me a book named "Our Mathematical Universe".
2. I will book a flight for San Francisco really soon!

The word *book* in the first sentence is clearly in a different sense from the second one. The first one denotes the "bound collection of pages" sense, while the other one bears the "to make an action or event a matter of record" sense.

Word sense disambiguation systems rely on knowledge resources. Resources are categorized into two: structured resources and unstructured ones. Structured resources are thesauri such as Roget's International Thesaurus (Chapman et al. (1984)), machine-readable dictionaries such as Collins English Dictionary or Oxford Dictionary of English, and ontologies such as WordNet (Fellbaum (1998)), Omega Ontology (Philpot et al. (2005)), OntoNotes (Hovy et al. (2006)) or BabelNet (Navigli & Ponzetto (2012)). Unstructured resources can be raw corpora such as Brown Corpus (Francis & Kucera (1979)) and uKWaC (Ferraresi et al. (2008)), sense-annotated corpora such as SemCor (Mihalcea (1998)), and collocation resources such as Word Sketch Engine¹ and JustTheWord².

Creating these resources is a highly labor-intensive and expensive task. Because of these drawbacks, only a few languages have similar resources. Moreover, natural languages are not stationary; these resources should periodically be updated, which makes the problem much more difficult. Also, the domain is another parameter to consider while creating these knowledge resources, and the process should be done for each domain separately because the sense inventory of the one domain might be radically different from the other. The necessity of this repetition is called *language acquisition bottleneck* (Gale et al. (1992)). On the other hand, Word Sense Induction (WSI) aims to *discriminate* (Schütze (1998)) different senses of a word from a corpus by unsupervised learning approaches without using any hand-crafted resources or manually-annotated data.

As mentioned above, there exist several drawbacks of representing the word senses with a fixed list of definitions of a manually constructed lexical database. There is no guarantee that they reflect the exact meaning of a target word in a

¹<https://www.sketchengine.co.uk/>

²<http://www.just-the-word.com/>

given context, since they usually contain definitions that are too general (Véronis (2004)). More so, lexical databases often include many rare senses while missing corpus/domain-specific senses (Pantel & Lin (2002)). Word Sense Induction (WSI) focuses on discriminating the usages of a polysemous word without using a fixed list of definitions or any hand-crafted resources.

1.2 Thesis

This thesis adopts and extends the pipeline for Unsupervised Part-of-Speech Tagging proposed by Yatbaz et al. (2012) for Word Sense Induction (WSI). Our method obtains a probability distribution for each context suggested by a statistical model. These substitute word distributions help to create context embeddings using the co-occurrence framework proposed by Globerson et al. (2004). Their method model the co-occurrence statistics by providing low-dimensional, dense vectors (embeddings) in Euclidean space. These embeddings can be word embeddings and context embeddings, depending the input data. Word embeddings can be transformed into context embeddings by using substitute word distribution obtained in the first step. Then, context embeddings are clustered by k -means clustering algorithm to distinguish usages (senses) of a word. I analyzed the method and its variations with three different lexical sample datasets. This thesis also provide comparisons with different word embeddings in literature along with the comparisons of different clustering algorithms and clustering approaches.

1.3 Motivation

Sense discovery is crucial for many problems in Natural Language Processing (NLP).

Machine Transition is a system responsible for translating one language to another. Identifying the correct sense of a word enables a system to correctly translate a sentence.

Lexicography is the discipline to compile, evaluate, and design dictionaries. The language is not stationary, and identifying new usages of words, or detecting emerging senses, can help in this area.

Question Answering is a discipline to answer questions asked by humans. A decent system needs to retrieve correct content, even if not a single word is similar with the question.

Information Retrieval is a discipline to present the most relevant information according to unstructured text. Schütze & Pedersen (1995) showed that using a highly accurate word sense disambiguation system increases the accuracy of an Information Retrieval system from 29.9% to 34.2%.

1.4 Outline

The rest of the thesis is organized as follows:

Chapter 2 is devoted to an overview of the related work and the different Word Sense Induction techniques used in literature.

Chapter 3 overviews the supervised and unsupervised evaluation methodologies commonly used in WSI tasks.

Chapter 4 is dedicated to explaining the proposed embedding-based system for the problem of sense discovery.

Chapter 5 details the variations of proposed systems. This chapter contains analysis of different context representations, different clustering approaches and different clustering algorithms. Three different lexical sample datasets are used for the experiments. Each experiment focuses the different part of the system and helps us to pick right combination for the final system. The chapter concludes by comparing our final system with the different WSI systems in literature.

Chapter 6 concludes the work in this thesis and suggests future topics and questions for investigation.

Appendix contains the important information about the datasets used in this thesis.

Chapter 2

Related Work

In this chapter, I will summarize the different representations and techniques for Word Sense Induction.

2.1 Representations

The frequency is used to represent queries and documents as vector was introduced by Salton et al. (1975). Vector Space Models (Turney & Pantel (2010)) gained a lot of attention for semantic information they provide. The distributional semantics hypothesis says that words observing in the same context, have similar meanings. This idea is popularized by the phrase “You shall know the word by company it keeps.” (Firth (1957)). Using a frequency matrix where rows and columns are words, cells denote how many times two words co-occur together (i.e., how many times word i appears with word j in a particular context, such as a context window, sentence, paragraph, document). Since raw counts of words that are occur in the same context may not be informative, there are methods to convert the frequency matrix into tf-idf (term frequency - inverted document frequency) (Sparck Jones (1972)) or Pointwise Mutual Information (PMI, Church & Hanks (1990)) counterparts. One of the extensive works on first-order and the second-order features was proposed by Schütze (1998). The first-order features denote the immediate neighbors of a target word—that is, the set of words that co-occur with a target

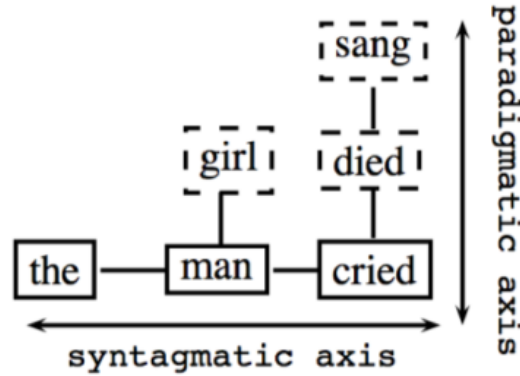


Figure 2.1: Syntactic and paradigmatic axes of *the man cried* (Chandler (2007))

word. On the other hand, second-order features (second-order co-occurrences) can be thought of as friends-of-friends: The set of words co-occur with not only target word but also co-occurring words with the target word. Purandare & Petersen (2004) analyzes word sense discrimination techniques that are used in first-order and second-order features.

Representing words as low-dimensional, dense vectors is increasing popularity for many problems in NLP. There are two ways of obtaining word embeddings. The first one which is based on representation of context as a term (token) vector is Latent Semantic Analysis (LSA) (Deerwester et al. (1990); Landauer & Dumais (1997)). LSA uses term-document (context) matrix where each cell contains the frequency of each term in a particular document. After computing these frequencies, singular value decomposition (SVD) is employed. This method provides three matrices and the important latent dimensions, in order. Using k latent dimensions, the original frequency matrix is mapped into a denser and lower-dimensional counterpart. This enables computational efficiency and helps to combat the sparsity of the original matrix. Van de Cruys & Apidianaki (2011) proposed a unified model based on LDA, and the system outperformed the systems participating in the SemEval-2010 word-sense induction and disambiguation tasks.

The second one is based on neural networks. Recently, there is a high volume of methods proposed that utilize neural networks to obtain word embeddings. For example, Collobert & Weston (2008) proposed a unified model based on convolutional neural networks, which predict (1) part-of-speech tags, (2) chunks, (3) named-entity tags using labeled data, and (4) likelihood of a sentence, (5) representation of a word type using unlabeled data. Mnih & Hinton (2009) presents a hierarchical language model based on Morin & Bengio (2005), which learns to represent words. Feedforward neural network language models have certain limitations such as the requirement to specify the context length. Mikolov et al. (2010) used *simple recurrent neural networks* to overcome this problem and showed a significant performance improvement over state-of-the-art backoff models. As a byproduct, the neural network also learns word representations. These word embeddings contain syntactic and semantic information (Mikolov, Yih, & Zweig (2013)). Mikolov, Chen, et al. (2013) proposed efficient architectures to compute continuous word representations. Globerson et al. (2004) described another method called CODE for embedding objects such as word in a single Euclidean space based on their co-occurrence statistics. Spherical CODE (S-CODE, Maron et al. (2010)) constrained the CODE method so that embeddings lie on a high-dimensional unit sphere, which provided an efficient optimization on large datasets and high-dimensional embeddings. Yatbaz et al. (2012) extended this method using most-probable substitutes (paradigmatic relations; see Figure 2.1) obtained by a language model and achieved best-known results on the unsupervised part-of-speech induction problem. A similar method is used in Baskaya et al. (2013) with a crucial difference: Context is represented by most-probable substitutes of each target word, and their clusters defined the sense of the target word. This difference enables one to identify the different usages of a polysemous word. Cirik & Yuret (2014) investigated the substitute-based S-CODE word embeddings on supervised tasks such as Named Entity Recognition (NER), Chunking, and Dependency Parsing. Yatbaz et al. (2014) proposed a method which solved the limitation of the previous method (Yatbaz et al. (2012) does represent the word types; however, words can have more than one part-of-speech, and context is important when deciding the part-of-speech

of a word) by representing not only the word but also the context. This instance-based method was evaluated with 19 different corpora in 15 languages and achieved results which were significantly better than or comparable to the best published word- or instance-based systems. Sert (2013) presents a comprehensive description for word (type) and context (token) embeddings obtained by S-CODE.

2.2 Techniques

Word Sense Induction methods can be categorized as follows:

1. Clustering methods
2. Bayesian models
3. Graph-based techniques
4. Frequent-termsets based models

2.2.1 Clustering Models

Firstly, there are two approaches to extract senses of a word: global and local.

Global algorithms (also known as *word clustering algorithms*) try to discriminate word senses using the other words. For instance, the Clustering-by-Committee (CBC) algorithm proposed in Pantel & Lin (2002) starts by discovering a set of tight (unambiguous) clusters where each centroid of the clusters is called the *committee*. Remaining word types can be assigned to one of the closest committees, and the key idea here is that after a word is assigned to a committee, the feature vector of assigned word is subtracted from the feature vector of the committee. This may provide a way to discover less dominant senses of the words. On the other hand, local algorithms aim to discover senses of a word independently of the other words, representing the context for each occurrence and clustering those contexts (Schütze (1998)).

Local clustering algorithms (also known as *context-clustering algorithms*) represent the context of a word by co-occurrence statistics. Then clustering methods can be employed to those context vectors (see previous section for different representations). Each cluster denotes a different sense of the word in question.

2.2.2 Graph-based Models

Graph-based algorithms utilize co-occurrence statistics. Typically, the semantics of a word are represented by a co-occurrence graph whose vertices bear co-occurrence information and whose edges hold the co-occurrence statistics. Véronis (2004) proposed HyperLex, a graph-based algorithm, which uses all words (i.e., global approach) and also a specific graph property called "small-world", which means that graph has highly interconnected groups (hubs) unlike random graphs, and these hubs eventually denote the senses of a word. Agirre, Martínez, de Lacalle, & Soroa (2006) extended their previous work in Agirre, Martínez, de Lacalle, & Soroa (2006) by adapting the PageRank algorithm Page et al. (1999) to compare their system with HyperLex on the Senseval-3 Lexical sample and on all-words tasks. UoY, proposed in Korkontzelos & Manandhar (2010), is another graph-based algorithm that participated in the SemEval-2010 WSI task. UoY constructs a graph with units as its nodes. A unit can be either a single word if it is predicted as unambiguous or word groups, assuming that word groups represent only one sense of a word (Klapaftis & Manandhar (2008)). Then Chinese Whispers (Biemann (2006)) runs on the constructed graph. They achieved relatively high scores in the task. UoS, proposed in Hope & Keller (2013b), constructed a dependency-parsed graph using uKWaC, a web-based corpus. They used Pointwise Mutual Information measure (Bouma (2009)) while creating the graph. MaxMax (Hope & Keller (2013a)) is used to cluster the graph. Finally, they merged the clusters according to *cohesion* and *separation* (Pang-Ning et al. (2006)) measures. Jurgens (2011) used a graph approach posing the problem as a community-detection problem, a well-studied problem in network science. Since using co-occurrence information is prohibitively expensive, the author created a representative subset of the original.

Then a co-occurrence graph was built with an iterative process. Each co-occurrence of a term with context has an edge created if there is none; otherwise, the edge's weight is increased by one. Edges whose weights were below 25 were pruned. Then Link Clustering (Ahn et al. (2010)) was used on the graph for community detection and achieved competitive performance on the dataset provided in SemEval 2010 WSI task.

2.2.3 Bayesian Approaches

Brody & Lapata (2009) proposed a Bayesian approach to model the contexts of the ambiguous word as samples from a multinomial distribution over senses which are in turn characterized as distribution over words. Their work is related to latent Dirichlet allocation (LDA, Blei et al. (2003)). Difference is that while LDA generates word from *global topics*, they generate words from *local topics* which are chosen based on the words observed in a context window around the ambiguous word. Their work is also categorized as local approach because they built a model for each word in question. The model also considers the part-of-speech and dependency information. They outperformed the systems participated in SemEval 2007 sense induction and discrimination task (Agirre & Soroa (2007)). Yao & Van Durme (2011) proposed a nonparametric Bayesian Model, Hierarchical Dirichlet Allocation (HDP) for Word Sense Induction. The main drawback of LDA is that number of senses of each word in question is specified as a parameter beforehand. Nonparametric methods (Reisinger & Mooney (2010); Vlachos et al. (2009)), however, make this decision automatically. They evaluated their system using supervised framework proposed in Agirre & Soroa (2007), and provided a comparison with LDA based model suggested in Brody & Lapata (2009). The training data is British National Corpus Clear (1993) and the test data they used is SemEval-2007 WSI. Their model which identified sense granularity performed equivalent or better than Brody & Lapata (2009). Another HDP-based model is Lau et al. (2013), and achieved competitive performance in SemEval-2013 WSI task.

2.2.4 Frequent-termsets Models

Termsets can be defined as a set of words frequently co-occur together in a document. Frequent termsets mining aims to reveal patterns (termsets) that are observed frequently in data. First, Rybinski et al. (2008) proposed a text mining approach based on frequent termsets. This method is domain- and language-independent; however, it requires part-of-speech information. The method consists of determining atomic contexts of terms of interest by means of maximal frequent termsets, which then are used for determining discriminant contexts. Recently, SnS (Sense Search) was proposed by Kozłowski & Rybiński (2014). This algorithm takes a textual corpus, the target word, and similarity threshold values. First, it discovers contextual patterns from the context. Context consists of proper names, nouns, and named entities which come from a part-of-speech tagger and a named-entity recognizer. This pattern discovery is made by the CHARM algorithm (Zaki & Hsiao (2002)), an efficient algorithm for mining closed frequent itemsets. With these contextual patterns, SnS extracts sense frames, then clusters these sense frames.

Chapter 3

Evaluation Framework

There are four different approaches to evaluate a WSI system. One alternative is manual evaluation. In other words, deciding the correctness of induced senses for each occurrence of the word manually. As expected, this evaluation strategy is very hard, labour-intensive, and unscalable. Moreover, manual evaluation needs to be done for every run of the system, even for previously evaluated instances. A second alternative can be to use clustering metrics (which is called *unsupervised evaluation*) such as V-measure, F-Score, B-Cubed (more details: [Amigó et al. \(2009\)](#)). The third one utilizes a mapping procedure (which is called *supervised evaluation*) which transforms each induced sense to one of the senses in a given lexicon such as Wordnet ([Pantel & Lin \(2002\)](#); [Agirre, Martínez, de Lacalle, & Soroa \(2006\)](#)) and evaluates the system's F-Score with using mapped senses and gold standards. The third alternative is to use extrinsic evaluation (also called *evaluation in use*): Evaluate the system on a specific application such as machine translation. This approach seems to be appealing but it may not be feasible for making lots of experiments in the development stage of the system, since it requires considerable time and effort ([Agirre & Soroa \(2007\)](#)).

The second and the third approaches (especially the supervised evaluation) gained popularity with the SemEval 2007 Word Sense Induction task, and it was used in following Word Sense Induction and Disambiguation tasks. Unfortunately

both supervised and unsupervised metrics have degenerate cases. In SemEval 2010 WSI task, organizers picked Paired FScore, V-Measure and supervised evaluation methodology (i.e., Agirre’s method) for system evaluation. The organizers of SemEval 2013 WSI task chose to use Fuzzy Normalized Mutual Information (FNMI), Fuzzy B-Cubed (FBC) along with various other clustering metrics, and supervised evaluation. Recently, researchers report their results using FNMI and FBC metrics. I calculated the performance of the systems using FNMI, FBC and supervised evaluation.

In this chapter, I will explain the most common evaluation metrics. I will start by describing the unsupervised evaluation methodology with five different metrics: F-Score, Paired F-Score, V-Measure, Fuzzy Normalized Mutual Information, and Fuzzy B-Cubed. Afterwards I will continue with the commonly-used mapping procedure for supervised evaluation: Agirre’s method. Note that both supervised and unsupervised evaluation metrics use a gold standard. The former method is called supervised since it learns a mapping between induced senses and senses in the lexicon used for the test set. Section 3.2 describes supervised evaluation framework in detail.

3.1 Unsupervised Evaluation

This section describes five different clustering metrics and their weak points.

3.1.1 FScore

Fscore (also known as clustering accuracy) whose calculation is similar to the commonly-used Information Retrieval metric called F-Measure [van Rijsbergen \(1979\)](#) is one of the most popular metrics in this domain. Basically, precision is defined as the ratio of correctly tagged examples for a cluster to total cluster size, and recall is defined as the ratio of correctly tagged examples to total class size. The formulation of FScore is as follows:

Let N be the number of instances, G the set of classes (gold senses), C the set of clusters (induced senses) and n_{ij} be the number of instances that are tagged as $g_i \in G$ and elements of $c_j \in C$.

$$R(g_i, c_j) = \frac{n_{ij}}{|G_i|} \quad (3.1)$$

$$P(g_i, c_j) = \frac{n_{ij}}{|C_j|} \quad (3.2)$$

$$F(g_i, c_j) = \frac{2 * R(g_i, c_j) * P(g_i, c_j)}{R(g_i, c_j) + P(g_i, c_j)} \quad (3.3)$$

$$FScore(G, C) = \sum_{g_i \in G} \frac{n_{ij}}{N} \max_{c_i \in C} F(g_i, c_j) \quad (3.4)$$

The highest FScore, which is observed when the clustering solution and the original classes are identical, is 1. A higher FScore means the clustering is better. Since recall is calculated as a ratio of common items between class i and cluster j , it answers how complete the solution is, while precision is calculated as a proportion of items belonging to cluster j and also members of class i , thus answering how homogeneous the given cluster j is with respect to class i .

The crucial problem of this technique is the *problem of matching*, which is explained in Meilă (2003). While calculating the system performance this metric considers only the contributions of the clusters that are matched with a gold class. Thus, two significantly different systems can get identical scores. Agirre & Soroa (2007) noticed that FScore tends to penalize the systems that return a high number of different senses, and it favors the systems with a low number of induced senses.

3.1.2 Paired FScore

The second technique to calculate the performance of a WSI system is Paired FScore (Artiles, Amigó, & Gonzalo (2009)), which is based on a combinatorial

	S_1	S_2	S_2
K_1	20	5	3
K_2	5	50	0
K_2	1	7	60
K_2	5	0	0

Table 3.1: Contingency table (S denotes gold senses, K denotes clusters)

approach, similar to RandIndex (Rand (1971)). The idea is to use a binary combination of clusters and classes and to calculate the overlap. There are 4 different options for a cluster-id and class-id pair: A pair can be in

- the same cluster and the same class
- the same cluster but the different classes
- the different clusters and the same class
- the different clusters and the different classes

Based on these four values, different approaches are proposed in literature. One of them is Paired FScore and it is calculated as follows:

$$P = \frac{|F(C) \cap F(G)|}{|F(C)|} \quad (3.5)$$

$$R = \frac{|F(C) \cap F(G)|}{|F(G)|} \quad (3.6)$$

$$Paired FScore = \frac{2 * P * R}{P + R} \quad (3.7)$$

where $F(C)$ is the set of instance pairs consisting of the combination of each instance labeled as $c_i \in C$. $F(G)$ is the set of instance pairs consisting of the combination of each instance annotated as $g_j \in G$.

Apidianaki & Van de Cruys (2011) show that the paired FScore has two degenerate cases: a system that induces a higher number of clusters than the gold senses gets low recall, and a system that induces a lower number of clusters than number of gold senses gets low precision.

3.1.3 V-Measure

Rosenberg & Hirschberg (2007) claims that their proposed metric, V-Measure, solves many problems that previous methods had difficulties with, such as (1) dependence on data set or algorithm, (2) *matching-problem* where part of the clustering is ignored in evaluation, (3) not focusing on complementary aspects of the nature of clustering (completeness and homogeneity) in order to provide accurate evaluation.

V-Measure is an entropy-based measure that is calculated by the harmonic mean of its two criteria: homogeneity and completeness. Homogeneity answers the question of how strongly the clusters in clustering result are assigned to one type of class. On the other hand, completeness explains to what degree data points of each type of class are assigned to a single cluster.

For the discussion, let A be the contingency table where rows denote clusters (K) and columns denote the gold senses (S) and a_{ij} be the number of all data points tagged by K_i which are members of S_j . Let N denote the total number of data points for target word t .

Homogeneity:

$$h = \begin{cases} 1, & \text{if } H(S) = 0 \\ 1 - \frac{H(S|K)}{H(S)}, & \text{otherwise} \end{cases} \quad (3.8)$$

$$H(S) = - \sum_{i=1}^{|S|} \frac{\sum_{j=1}^{|K|} a_{ij}}{N} \log \frac{\sum_{j=1}^{|K|} a_{ij}}{N} \quad (3.9)$$

$$H(S|K) = - \sum_{i=1}^{|K|} \sum_{j=1}^{|S|} \frac{a_{ij}}{N} \log \frac{a_{ij}}{\sum_{s=1}^{|S|} a_{is}} \quad (3.10)$$

Completeness:

$$c = \begin{cases} 1, & \text{if } H(K) = 0 \\ 1 - \frac{H(K|S)}{H(K)}, & \text{otherwise} \end{cases} \quad (3.11)$$

$$H(K) = - \sum_{i=1}^{|K|} \frac{\sum_{j=1}^{|S|} a_{ij}}{N} \log \frac{\sum_{j=1}^{|S|} a_{ij}}{N} \quad (3.12)$$

$$H(K|S) = - \sum_{i=1}^{|S|} \sum_{j=1}^{|K|} \frac{a_{ij}}{N} \log \frac{a_{ij}}{\sum_{s=1}^{|K|} a_{is}} \quad (3.13)$$

Finally, V-Measure is defined as follows:

$$V\text{-Measure} = \frac{2 * h * c}{h + c} \quad (3.14)$$

For the contingency matrix in Table 3.1, homogeneity is equal to .596, completeness is equal to .547, and V-Measure is equal to .570¹.

Although V-Measure seems to be a robust measure, in the SemEval 2010 WSI task, Pedersen (2010) provided baselines with different levels of randomness and showed the trend that scores were increased with increased randomness of the baselines. In fact, the author's two random baselines (their V-Measure scores were 18.7, 23.2) outperformed the best participating system (its V-Measure score was 16.2).

3.1.4 Fuzzy B-Cubed

Amigó et al. (2009) proposed comprehensive extrinsic comparisons with the cluster-

¹V-Measure implementation can be found at <https://goo.gl/mmj97e>

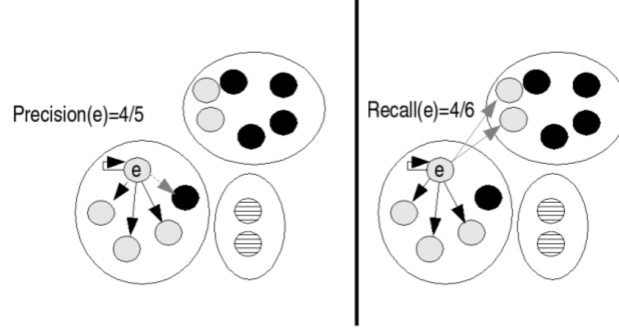


Figure 3.1: Precision and recall calculation for B-Cubed (Amigó et al. (2009)). Different fillings denote gold labels whereas big circles denote different clustering answers.

ing evaluation metrics according to four constraints they suggested: *homogeneity*, *completeness*, *rag bag*, which means “disorder into a disordered cluster is less harmful than introducing disorder into a clean cluster”, and *clusters size vs. quantity*, which means “A small error in a big cluster should be preferable to a large number of small errors in small clusters”. Their findings showed that only B-Cubed satisfies these constraints. B-Cubed is based on precision and recall, which estimate the fit between two clustering answers in item level, proposed by Bagga & Baldwin (1998). Figure 3.1 illustrates how precision and recall can be calculated for an item e . Let assume cluster label of e is C and its gold label is G . Precision is the ratio of total number of items in C whose gold label is G to size of C , and recall is the ratio of number of items in C whose gold label is G to size of G . The original version of B-Cubed metric does not handle the situation where one item is assigned to more than one clusters (which is called *overlapping clusters* or *covers*). Thus, Amigó et al. (2009) extended the definition of B-Cubed to allow it to compare overlapping clusters. Jurgens & Klapaftis (2013) generalizes B-Cubed in order to handle partial membership (e.g., a movie’s genre can be labeled 0.6 sci-fi, 0.2 adventure, 0.2 drama) of clustering answers, and they called it Fuzzy B-Cubed.

3.1.5 Fuzzy Normalized Mutual Information

Mutual information measures the dependence between two random variables. In the clustering evaluation context, clusters (induced senses) and sense classes (gold senses) are treated as random variables and mutual information computes the level of agreement between these labels. Since this measure is used for comparisons with multiple clustering solutions (WSI systems), it is normalized to $[0, 1]$ by $\max(H(X), H(Y))$, as recommended in Vinh et al. (2009). However, the mutual information is defined only to compare non-overlapping cluster partitions. Based on Lancichinetti et al. (2009), Jurgens & Klapaftis (2013) extended the work in order to provide mutual information between overlapping clusters.

3.2 Supervised Evaluation

3.2.1 Agirre’s method

Agirre, Martínez, ó de Lacalle, & Soroa (2006) proposed a supervised evaluation framework for WSI systems. For each target word t , this framework splits the the actual test set into two: the training part and the test part. In SemEval tasks, the same split is used for evaluating all the systems. The training part is used for automatically learning the mapping from induced senses to gold senses for t . This obtained mapping is used to evaluate the instances in the test part. This supervised evaluation was first used in the SemEval 2007 WSID task, and after that it became an important evaluation setting for consecutive SemEval WSI and WSD tasks. From my experiments, Agirre’s method also penalizes systems that provide a high number of clusters since this method is not able to map an induced sense that occurred in the test part but not observed in the training part. One improvement is that instead of dividing the data randomly, the division can aim to preserve the distribution of senses for both splits (e.g., Stratified K-Fold Validation). I used this methodology throught the thesis.

Chapter 4

Method

This chapter describes a proposed system for the WSI problem. The pipeline starts with calculating the substitute distribution of a word in a given context, explained in Section 4.2. Then, a list of substitutes is generated by using this substitute distribution. Each substitute word is paired with a context-id. A context-id (e.g., *car*₂) is a unique identifier that helps to distinguish each context from others. The Euclidean embedding algorithm S-CODE uses the co-occurrence data (context-id and substitute word pair) and outputs the embedding for each context. Section 4.4 clarifies the co-occurrence modeling step. When each context is represented in Euclidean space, the k -means algorithm is employed to cluster these context embeddings, will be detailed in Section 4.5.

4.1 Data Enrichment

Sense annotation is a hard task even for humans. For the lexical sample tasks, organizers provide a small test dataset annotated by humans, which is used for system evaluation. The systems should use training corpus, if provided, or corpora for sense induction. Typically test data does not contain enough instances for neither sense induction nor word sense disambiguation. Data enrichment aims to increase the number of instances of target words, explained as follows: Assuming

that our target word is *book* in noun form, I randomly fetch an arbitrary number of additional contexts from a large corpus where our target word occurs with the same part-of-speech tag. This implies that I skip those sentences in which the word *book* functions as a verb.

For all experiments in Chapter 5, the number of additional contexts is fixed and equal to 20000, and uKWaC is used as a corpus for data enrichment. In addition to enrichment for target words in test datasets, I fetch additional instances for verbs, nouns, and adjectives. The preliminary experiments show that this additional data helps to cover more words and their interactions in co-occurrence modeling step, it help to provide better embeddings.

4.2 Substitute Word Distributions

In this section, I will explain how substitute word distribution captures the paradigmatic relations of word tokens and how it is computed. Sahlgren (2006) investigates the syntagmatic and paradigmatic relations captured by the Vector Space Models. His paradigmatic representation models word types using co-occurrence counts. Following the methodology used in Yatbaz et al. (2012), I represent the context of word tokens using probabilities of likely substitutes. The substitute word distribution is a probability distribution of observing a word in the context of a word token. The context of a target word in a sentence is defined as the sequence of words in the window of size $2n - 1$ centered at the position of the target word token. The context excludes the target word token. I will use the following example sentence to clarify the definition of context:

*“Accordingly, the last chapter of this **book** is devoted to our reflections on the impact of channel integration on certain public policies issues.”*

In the sentence above, the context of the word token ‘*book*’, when $n = 4$, is “*chapter of this — is devoted to*”, where ‘—’ denotes the target word token.

Assuming the position of the target word token is 0, the context spans from positions $-n + 1$ to $n - 1$, and the probability of observing each type w in our vocabulary in the context of the target word token is computed using the following equation:

$$P(w_0 = w | c_{w_0}) \propto P(w_{-n+1} \dots w_0 \dots w_{n-1}) \quad (4.1)$$

$$= P(w_{-n+1}) \dots P(w_0 | w_{-n+1}^{-1}) \dots P(w_{n-1} | w_{-n+1}^{n-2}) \quad (4.2)$$

$$\approx P(w_{-n+1}) \dots P(w_0 | w_{-n+1}^{-1}) \dots P(w_{n-1} | w_0^{n-2}) \quad (4.3)$$

$$\propto P(w_0 | w_{-n+1}^{-1}) \dots P(w_{n-1} | w_0^{n-2}) \quad (4.4)$$

Where w_i^j is the word sequence from w_i to w_j (for $i < j$) and c_{w_0} is the context of the target word token at position 0 of length $2n - 1$, meaning w_{-n+1}^{-1} to the left and w_1^{n-1} to the right of w_0 .

In the Equation 4.1, the right-hand side is proportional to the left-hand side because $P(c_{w_0})$ is independent of any given word w for w_0 . Using the chain rule, Equation 4.2 is obtained from the Equation 4.1. By applying the n^{th} -order Markov assumption, only the closest $n - 1$ words in each term of the Equation 4.2 are used, and others are excluded. The result is the Equation 4.3. The Equation 4.4 is proportional to the Equation 4.3 because terms which do not depend on w_0 are fixed, since the context of the word at position 0 is fixed. Near the boundaries of the sentence, specifically the first and last $n - 1$ words, the appropriate terms of the Equation 4.4 are truncated or dropped (e.g. if 0 is the first word of a sentence, $P(w_0 | w_{-n+1}^{-1})$ becomes $P(w_0)$).

The probabilities required to compute the Equation 4.4 can be obtained from an n -gram language model. For this step, I used SRILM (Stolcke (2002)) to build a 4-gram language model with interpolated Kneser-Ney discounting using uKWaC, a web corpus with 2 billion tokens. Words that were observed less than 2 times in the language model training data were replaced by `<unk>` tags, which gave us a vocabulary size of 4,254,946. For computational efficiency, only the top 100

Substitute Word	Probability
book	.638
section	.139
paper	.089
volume	.053
document	.014
report	.013
work	.012
tutorial	.005
edition	.003
biography	.001

Table 4.1: Top 10 substitutes and their probabilities for the context, “*chapter of this — is devoted to*”.

substitutes and their probabilities were computed for each position in the context of a target word using the FASTSUBS algorithm (Yuret (2012)). These top 100 probabilities were normalized to add up to 1.0, giving us a legitimate probability distribution for a context. As you can see in Table 4.1, the high-probability substitutes reflect both semantic and syntactic properties of the context of a target word in the sentence above.

4.3 Discretization of the Substitute Word Distribution

Although the context of a target word was modeled as a substitute word distribution, the Euclidean co-occurrence modeling algorithm which will be introduced in the next section expects its input to be discrete objects. I will list two methods to discretize substitute word distribution into its discrete counterpart. Sert (2013) proposed a method which creates a proximity matrix of substitute word distributions and computes the ids of the k -nearest neighbors of each distribution. The

Instance ID	Substitute Word
book.n.1	book
book.n.1	book
book.n.1	section
book.n.1	book
book.n.1	book
book.n.1	paper
book.n.1	book
book.n.1	volume
...	...
book.n.1	book
book.n.1	document

Table 4.2: An example output of discretization of the substitute word distribution which is illustrated in Table 4.1. *book.n.1* is the instance id of the target word *book* that functions as noun.

second method introduced in [Yatbaz et al. \(2012\)](#) draws k word tokens with replacement from the substitute word distributions. This method is computationally cheaper than the former one; however, the number of samples should not be small and requires attention to pick. Too small k values make this setting not represent the distribution, and too large k values make the following steps computationally expensive. From my preliminary experiments, 100 samples was enough for each context. Thus, for each context, I sampled 100 word tokens from their corresponding substitute distributions. Table 4.2 illustrates the output of discretization process of the substitute word distribution which is illustrated in Table 4.1. Note that this example shows discretization of only one substitute word distribution. In co-occurrence modeling step, we will first group discrete versions of word distributions according to the part-of-speech tag of target words, and then employ the co-occurrence modeling algorithm for part-of-speech groups independently with each other.

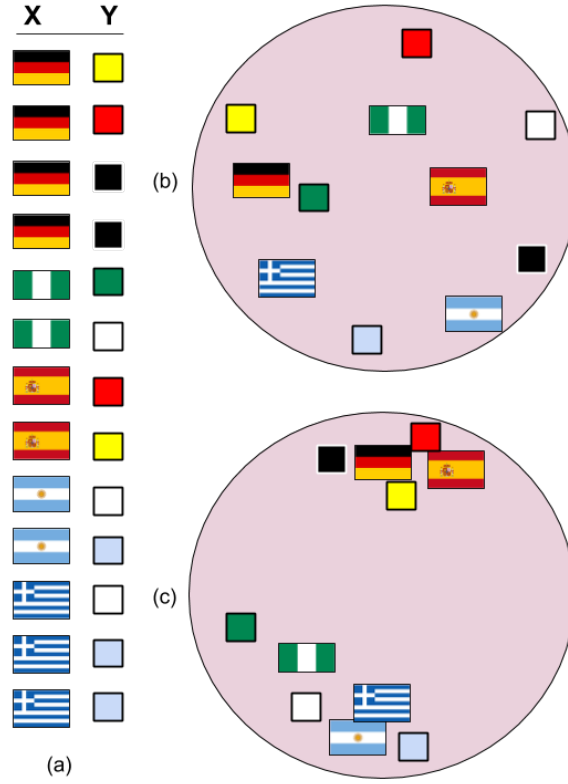


Figure 4.1: Figure 4.1.a illustrates an artificial co-occurrence dataset. X variable generates country flags, and Y variable generates colors. In Figure 4.1.b, S-CODE randomly maps all the values onto the surface of a d -dimensional sphere. Figure 4.1.c depicts that S-CODE runs and converges. Embeddings of pairs co-occurring together will be located closely on the unit sphere.

4.4 Co-occurrence Modeling with S-CODE algorithm

In this section, I give a brief introduction for the Symmetric Interaction Model of the Co-occurrence Data Embedding (CODE, Globerson et al. (2004)) and its extension, Spherical Co-Occurrence Data Embedding (S-CODE, Maron et al. (2010)). S-CODE randomly initializes co-occurrence data generated using substitute word distributions and the discretization operation, explained in the previous section, into a d -dimensional Euclidean sphere. After the algorithm converges, the contexts with similar substitute word distributions are close to each other. In Section 4.5,

I will explain how S-CODE output is used and how it can be combined with substitute word distributions in order to obtain high quality context embeddings.

Let X and Y be two categorical variables with finite cardinality $|X|$ and $|Y|$. We observe a set of pairs $\{x_i, y_i\}_{i=1}^n$ drawn IID from the joint distribution of X and Y . These pairs are summarized by the empirical distributions $\bar{p}(x, y)$, $\bar{p}(x)$ and $\bar{p}(y)$. The task is to find embeddings $\phi(x)$ and $\psi(y)$ for each unique variable x and y . These embeddings need to reflect the statistical relationship between the variables x and y such that pairs occurring together frequently should be close to each other. Hence, the square of the Euclidean distance, $d_{x,y}^2 = |\phi(x) - \psi(y)|^2$, should be small in d dimensional space.

Later, Maron et al. (2010) constrain the embedding to lie on a high-dimensional unit sphere, which enables us to approximate the partition function, which is a dynamic variable, Z , using a constant, $\bar{Z}$, which arises from a coarse approximation in which all pairs of embedded variables are distributed uniformly and independently on the sphere. This constraint allows the algorithm to work on large datasets by not having to calculate Z after each gradient step. Thus, in this thesis I used Spherical CODE:

$$p(x, y) = \frac{1}{Z} \bar{p}(x) \bar{p}(y) e^{-d_{x,y}^2} \quad (4.5)$$

where $Z = \sum_{x,y} \bar{p}(x) \bar{p}(y) e^{-d_{x,y}^2}$ is the normalization term. We can express the log-likelihood of the joint distribution over all embeddings ϕ and ψ as the following:

$$\begin{aligned} \ell(\phi, \psi) &= \sum_{x,y} \bar{p}(x, y) \log p(x, y) \\ &= \sum_{x,y} \bar{p}(x, y) (-\log Z + \log \bar{p}(x) \bar{p}(y) - d_{x,y}^2) \\ &= -\log Z + \text{const} - \sum_{x,y} \bar{p}(x, y) d_{x,y}^2 \end{aligned} \quad (4.6)$$

The gradient of the log-likelihood depends on the sum of embeddings $\phi(x)$ and $\psi(y)$, for $x \in X$ and $y \in Y$, and to maximize the log-likelihood, Maron et al. (2010) use a gradient-ascent approach. The gradient is as follows:

$$\frac{\partial \ell(\phi, \psi)}{\partial \phi(x)} = \sum_y 2\bar{p}(x, y)[\psi(y) - \phi(x)] + \frac{1}{Z} \sum_y \bar{p}(x)\bar{p}(y)[\phi(x) - \psi(y)]e^{-d_{x,y}^2} \quad (4.7)$$

$$\frac{\partial \ell(\phi, \psi)}{\partial \psi(y)} = \sum_x 2\bar{p}(x, y)[\phi(x) - \psi(y)] + \frac{1}{Z} \sum_x \bar{p}(x)\bar{p}(y)[\psi(y) - \phi(x)]e^{-d_{x,y}^2} \quad (4.8)$$

The first sum, the gradient of the part with $d_{x,y}^2$ in (4.6), in (4.7) [(4.8)] acts as an attraction force between the $\phi(x)$ ($\psi(y)$) and all the embeddings ψ (ϕ), in proportion to respective joint empirical probabilities $\bar{p}(x, y)$. The second sum, the gradient of $-\log Z$ in (4.6), in (4.7) [(4.8)] acts a repulsion force between the $\phi(x)$ ($\psi(y)$) and all the embeddings ψ (ϕ), in proportion to respective marginal empirical probabilities $\bar{p}(x)$ and $\bar{p}(y)$.

Figure 4.1 illustrates the S-CODE algorithm on an artificial data. In this work, I use S-CODE with sampling based stochastic gradient ascent, a constant approximation of Z (which is 0.166) and randomly initialized ϕ and ψ vectors in 100 dimensional sphere ($d = 100$). The algorithm stops if the difference between two consecutive likelihood values is less than 0.0001.

I run S-CODE algorithm with two different inputs: First one is to use the all enriched data. I called this approach *global co-occurrence modeling*. The second one is called *pos-based co-occurrence modeling* where the enriched data was divided into three (noun, verb, adjective) according to the part-of-speech information of each word token in observing context. In other words, we have three S-CODE algorithm runs for *pos-based co-occurrence modeling*, each of which for a different part-of-speech (nouns, adjectives, verbs). The second approach provided significantly better embeddings, thus, I used *pos-based co-occurrence modeling* throughout this

thesis.

4.5 Clustering

The output of the previous step is context embeddings, which are used in clustering step. In this thesis, I tested three different clustering algorithms: Weighted k -means which is a k -means algorithm with (optional) instance weights and careful initialization (Arthur & Vassilvitskii (2007))¹. In the literature, there are two common methods to specify k , required parameter for k -means algorithm: *Gap Statistics* (Tibshirani et al. (2000)) and Pham’s heuristic (Pham et al. (2005)). In this thesis, I did not use any of these two methods; I report upper bound performance of our method by trying all plausible k values, and picking the best performing one in a particular test set.

I also tried Spectral Clustering (Shi & Malik (2000)) and DBSCAN (Ester et al. (1996)). The representations I used for clustering will be explained in the following section 5.2 and the performance of these algorithms will be compared in Chapter 5.

¹An efficient implementation can be found at <https://github.com/ai-ku/wkmeans>

Chapter 5

Experiments

In the previous chapter I explained the pipeline of operations I used for Word Sense Induction problem. In this chapter, I will introduce baselines for lexical sample tasks. Then, I will explain the different approaches in context representation and present a comparison of our context embeddings and different word embeddings in literature. Then, I describe different clustering approaches and algorithms. Next, I will provide a performance comparison with different systems in literature. For these comparisons I used three lexical sample datasets: (1) SemEval 2010 WSI task (details: [Appendix A.1](#)), (2) SemEval 2013 WSI task (details: [Appendix A.2](#)) and (3) the dataset I created using OntoNotes project (details: [Appendix A.3](#)).

5.1 Baselines

In this section, I will explain the Most Frequent Sense Baseline and the various random baselines for lexical sample task.

Most Frequent Sense Baseline (MFS): MFS assigns the most frequent sense in the test set to each instance of a word. For instance, assume that *car* has 3 different senses in the test set, and sense *car*₁ is the most frequent sense with 80%. MFS labels all instances of *car* with *car*₁.

Dataset	RB type	1 of 2	1 of 3	1 of n
SemEval 2010	Induced RB	0.585	0.575	0.574
	RB (uniform)	-	-	0.278
	RB (weighted)	-	-	0.494
SemEval 2013	Induced RB	0.562	0.555	0.533
	RB (uniform)	-	-	0.197
	RB (weighted)	-	-	0.451
OntoNotes	Induced RB	70.6	70.5	70.3
	RB (uniform)	-	-	24.4
	RB (weighted)	-	-	61.4

Table 5.1: Random baselines on SemEval-2010, SemEval 2013 and the OntoNotes dataset. For the induced baseline, contexts were randomly tagged, and afterwards Agirre’s mapping was applied. For each instance, the uniform baseline randomly picks one of the senses in the sense inventory with equal chance. The weighted one considers the sense distribution in the sense inventory.

Random Baselines (RBs): I calculated three types of random baselines: (1) a random induced sense baseline, (2) a uniform random baseline, and (3) a random baseline using the same sense distribution as the gold standard. The first one is a dummy system that provides random induced senses for each instance. Since this random baseline provides arbitrary senses (i.e., the sense inventory is not the same as the gold standard sense inventory), the mapping between these induced senses and gold standard senses needs to be done. The number in the name of the baseline indicates how many different induced senses were provided for each target word. The second type of random baseline uses the correct number of senses for each target word and picks a sense among those senses provided in gold standard with equal probability. The third one picks senses according to the sense distribution of each word. Table 5.1 shows the performance of various random baselines for SemEval 2010, SemEval 2013, and the dataset we created using OntoNotes 5.0 sense annotations, respectively.

5.2 Different Representations

In this section, I will begin by describing the methodology to transform word embeddings to context embeddings. Then, I will present the performance of our context embeddings and the context embeddings we created from word embeddings in literature. After that, I will present the results of our systems in three different lexical sample datasets, and compare our system performance with other systems participated in SemEval 2010, and SemEval 2013 tasks.

5.2.1 From Word Representation to Context Representation

In this section I will explain how to transform word embeddings to context embeddings. The co-occurrence modeling step explained in Section 4.4 provides two types of embeddings: context embeddings (from the first categorical variable) and word embeddings (from the second categorical variable). Recall that a word embedding is the representation of a word and it doesn't depend on the context. In order to distinguish the usages of a word you have to consider the context where that particular word occurs. By using the substitute word distributions, and the word embeddings obtained by co-occurrence modeling, one can transform word embeddings into context embeddings explained below.

Let E^K be a word embeddings K and E_w^K is the embedding of a word w in $\mathbb{R}^d$. If K does not contain w , then E_w^K returns $\mathbf{0}$ in $\mathbb{R}^d$. Let S be set of substitute word distributions and S_i is the substitute word distribution for context i , and S_i^w is the probability of observing a word w in the context i . Finally, the equation 5.1 provides context embedding using embedding set E , weighted by substitute word distribution of a given context i :

$$e_i = \sum_{w \in V} S_i^w E_w^K \quad (5.1)$$

where V denotes the vocabulary. This is *expected embedding over substitute*

word distribution (EESD). Note that there are many word embeddings in literature, and the equation above is applicable for those embedding sets.

The following lists the S-CODE embeddings used in the thesis:

1. *Baskaya* context embeddings
2. $SCODE_{EESD}$ context embeddings

The first embedding is obtained by *pos-based co-occurrence modeling* explained in Section 4.4. Important to note that *Baskaya* embeddings are context embeddings so it is unnecessary and inconvenient to apply Equation 5.1, which is only applicable for word embeddings. On the other hand, the other embedding set is the production of the Equation 5.1. Both use the same substitute word distribution, however, $SCODE_{EESD}$ is obtained by applying Equation 5.1.

5.2.2 Comparison of Different Context Representations

Various methods and word embeddings proposed in literature (see 2.1 for details). In the thesis, I created a benchmark for our context embeddings (*Baskaya* and $SCODE_{EESD}$) as well as the following word embeddings in the literature: C&W (Collobert & Weston (2008)), HLBL (Mnih & Hinton (2009)), GloVe (Pennington et al. (2014)), GC (Huang et al. (2012)), SkipGram (Mikolov, Chen, et al. (2013)), *Skip_{dep}* (Bansal et al. (2014)), and ML (Faruqui & Dyer (2014)). Before the comparison, these word embeddings are transformed to context embeddings using the substitute word distribution of each context according to Equation 5.1. This is the exact procedure that I followed in order to create $SCODE_{EESD}$ context embeddings. EESD stands for *expected embedding over substitute distribution*. Note that in the next section I will omit using *EESD* subscript for context embeddings which were first transformed from word embeddings by using the equation in the previous section and the substitute distributions, however, all comparisons were done over context embeddings.

Representation	# of word embeddings	Coverage (%)	Accuracy
SCODE_{EESD}	254746	100	74.301
Baskaya	254746	100	73.365
Skipgram	1119721	96.91	72.912
Glove	400001	92.49	71.160
Multilingual	180835	91.14	71.100
Skip_{dep}	316756	96.54	70.826
Global Context	100232	91.22	70.362
CW	268810	97.03	70.323
HLBR	246122	96.47	72.678
SubDist	-	-	68.118

Table 5.2: A comparison of the accuracies for various context embeddings and substitute word distributions (the last row) on SemEval 2010 WSI task. SVM with RBF kernel is used for all representations. The best parameter setting for each model is found using grid search. Accuracy was reported according to the Agirre’s method (supervised evaluation framework).

Comparison Methodology

Comparison methodology can be explained as follows. For each target word in the test data I trained Support Vector Machines [Drucker et al. \(1999\)](#) with an RBF kernel. Hyperparameters were optimized by grid search, and scores were obtained by 5-fold cross validation. I repeated these steps for each context embeddings.

Performance comparison of context embeddings in SemEval 2010

Table 5.2 summarizes each embedding’s performance and its coverage for the SemEval 2010 test data¹. Instead of using weighted average of a word embedding, one can use substitute word distribution directly, and the last row of the table shows the performance of direct usage of substitute word distributions of each context. According to this table, our context embeddings (first two rows) outperformed all

¹A detailed version (the performance of each representation, its performance on each target word and the best hyperparameter set for each model) can be found at <https://goo.gl/Zofhfm>

Representation	# of word embeddings	Coverage (%)	Accuracy (%)
SCODE_{EESD}	254746	100	69.502
Baskaya	254746	100	68.852
Skipgram	1119721	96.91	69.119
Glove	400001	92.49	68.529
Multilingual	180835	91.14	67.490
Skip_{dep}	316756	96.54	67.844
Global Context	100232	91.22	67.815
CW	268810	97.03	67.911
HLBR	246122	96.47	68.055
SubDist	-	-	65.542

Table 5.3: A comparison of the accuracies for various context embeddings and substitute word distributions (the last row) on SemEval 2013 WSI task. SVM with RBF kernel is used for all representations. The best parameter setting for each model is found using grid search. Accuracy was reported according to the Agirre’s method (supervised evaluation framework).

other context embeddings.

Performance comparison of context embeddings in SemEval 2013

Table 5.3 summarizes each embedding’s performance and its coverage for the test data¹. Context embeddings performed similarly and they outperformed *SubDist*. *SCODE_{EESD}* performed slightly better than *Skipgram*.

5.3 Variations in the Clustering step

In this section I will explain the different clustering approaches and different clustering algorithms in clustering step.

¹A detailed performance table can be seen at <https://goo.gl/010sp5>.

Dataset	Approach	Baskaya	SCODE _{EESD}
SemEval 2010	local	65.8 (k=10)	65.7 (k=10)
	pos-based	61.3 (k=72)	61.4 (k=32)
SemEval 2013	local	61.8 (k=6)	61.7 (k=6)
	pos-based	58.6 (k=4)	59.1 (k=8)
OntoNotes	local	74.5 (k=22)	74.6 (k=22)
	pos-based	72.8 (k=25)	72.9 (k=16)

Table 5.4: Performances of different clustering approaches on SemEval 2010, SemEval 2013, and OntoNotes dataset. Scores are calculated by supervised evaluation (Agirre’s method)

5.3.1 Different Clustering Approaches

The two clustering approaches tested in the thesis:

Local approach: In the local approach, clustering is employed for each target word, separately. For instance, if the lexical sample test data contains 50 target words, then clustering step runs for 50 times using only the instances of a particular target word in the test data and the enriched data. Assume that the target word is *book*. It has 100 instances in the test data. The input of the clustering algorithm is 20100 vectors, 100 from the test data, 20000 from the UKWaC. Note that this approach is prohibitively expensive for an all-words task.

Pos-based approach: In the lexical sampling task, test data contains the part-of-speech information for each target word. Recall that we fetched additional instances for each part-of-speech (noun, verb, adjective) we are interested in. (see Section 4.1). In the pos-based approach, clustering is applied instances of each part-of-speech, separately—that is, first instances are separated according to target words’ part-of-speech information, then the clustering algorithm applies on each subset of the data. This is more coarse-grained approach than the local approach; induced senses are more like semantic classes.

Dataset	Algorithm	Baskaya	SCODE _{EESD}
SemEval 2010	k-means	65.8	65.7
	Spectral Clustering	64.2	64.0
	DBSCAN	61.2	59.8
SemEval 2013	k-means	61.8	61.7
	Spectral Clustering	59.9	61.2
	DBSCAN	57.9	57.4

Table 5.5: Best scores of different clustering algorithms in SemEval 2010 and SemEval 2013 WSI Task. Scores are calculated by supervised evaluation (Agirre’s method)

Table 5.4 provides the performances of both clustering approaches on three different datasets. Local clustering approach outperforms the pos-based counterpart on all datasets for both context representations. Therefore, I picked local clustering approach for subsequent experiments. Note that pos-based clustering approach is faster than the local one. However, memory consumption is higher since it uses all target words with the same part-of-speech.

5.3.2 Different Clustering Algorithms

I compared three different clustering algorithms, k -means, Spectral Clustering and DBSCAN. The experiment in the previous section showed that local approach outperforms pos-based approach. Therefore, I compared three algorithms only in local setting. Hyperparameters of the Spectral Clustering and DBSCAN were optimized by hyperopt¹. For Spectral Clustering, parameters that I optimized are k (number of clusters), and affinity (nearest neighbors vs rbf). For DBSCAN, two parameters (eps and minimum sample size) were optimized. Table 5.5 depicts the best scores for these three algorithms on SemEval 2010 and SemEval 2013 tasks.

For both datasets k -means performed better than other two clustering algo-

¹<https://github.com/hyperopt/hyperopt>

System	SE-F1
MFS	58.7
RB (weighted)	49.4
UoY	62.4
Duluth-WSI	60.5
KCDC-PC2	59.8
Hermit	58.3
KSU KDD	52.2
SCODE _{EESD} (k=10)	65.7
Baskaya (k=10)	65.8

Table 5.6: A comparison of the accuracies with various systems. SE-F1score stands for the F1-Score of supervised evaluation using Agirre’s method. The top part of the table shows the baselines and the second part of the table illustrates the best systems of each participated team. The bottom part of the table contains the performance of our different word context representations in a local clustering setting.

rithms. Moreover, Spectral Clustering and DBSCAN are really slow when compared with k -means.¹.

5.4 Comparison between Our System and Other Systems

From the previous experiments ((1) context representation, (2) clustering approach (local vs pos-based) and clustering algorithm (k -means vs Spectral Clustering vs DBSCAN) I picked k -means as clustering algorithm and *local clustering approach* as the clustering approach for our final system. Since the performances of *Baskaya* and *SCODE_{EESD}* context embeddings are very close, I kept these context representations for the comparisons with other systems, which will be described in this section.

¹ <https://goo.gl/qO0SH7>

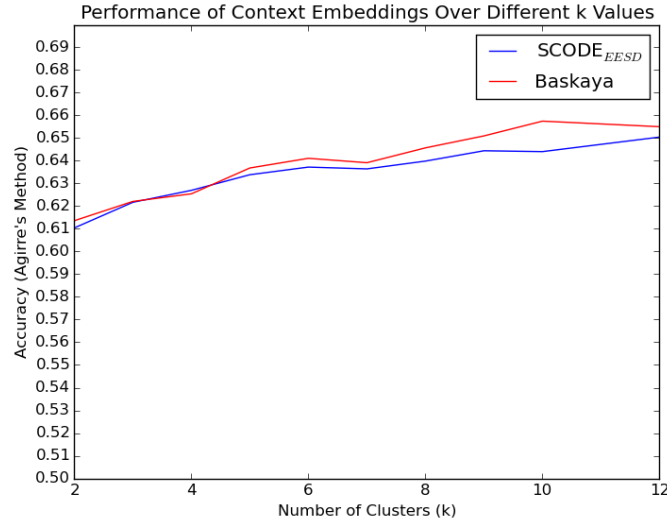


Figure 5.1: Performance analysis of S-CODE context embeddings with respect to the number of clusters in local clustering setting (Agirre’s method is used for evaluation).

5.4.1 SemEval 2010 WSI task

Figure 5.1 depicts the performance analysis of our context embeddings over different cluster sizes by using supervised evaluation framework. Two context embeddings perform closely in local clustering setting. Recall that local clustering means that clustering algorithm is employed each target word separately. The difference between the best and the worst score is $\sim 4\%$. Figure 5.2 shows the performance of two embeddings over cluster size with different unsupervised evaluation metrics.

There were 26 systems participating in this task. According to the supervised evaluation, 14 systems outperformed the MFS baseline. *UoY* (Korkontzelos & Manandhar (2010)) is the best-performing system in nouns and overall. The second part of the Table 5.6 contains participating team’s best systems according to the supervised evaluation results that the organizers provided. The third part of the table shows the score of our system with two context representations, which are explained in Section 5.2. I used k -means as a clustering algorithm. Note that these scores are an upper-bound of our systems. In other words, for each representation, I chose the best-performing k for the k -means algorithm by running the algorithm

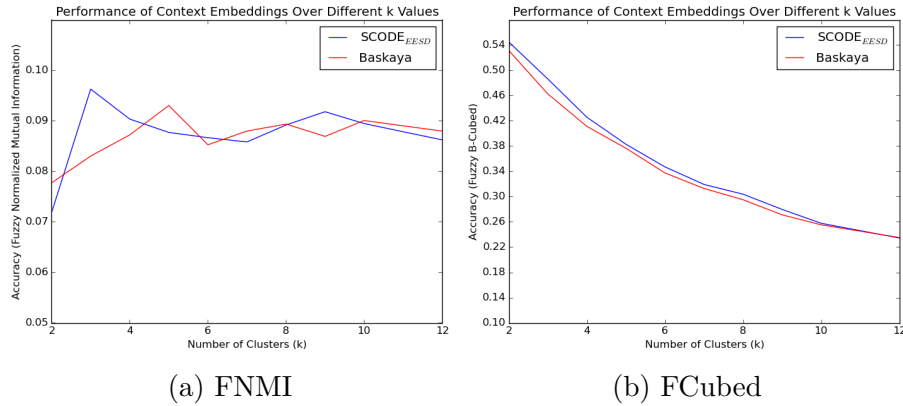


Figure 5.2: Performance differences of S-CODE context embeddings with different number of clusters in local clustering setting for SemEval 2010 WSI task. Accuracies are computed by using (a) Fuzzy Normalized Mutual Information and (b) FCubed.

with different k values. The proposed embeddings outperformed all the baselines and the best participating system in SemEval 2010. The embeddings in the pos-based clustering setting also performed well although the nature of the supervised evaluation metric is to punish systems that return a high number of induced senses. However, the systems outperformed both MFS (58.7%) and random baselines.

5.4.2 SemEval 2013 WSI task

In this section, I will describe the systems that were participating in SemEval 2013 Task 13, along with recent systems applied to the same dataset. At the end, I will present a table that summarizes the performance of each system.

There were four participants for SemEval-2013 Task 13. Three of them were using induced sense inventory, and the remaining one was a WSD system.

AI-KU: Baskaya et al. (2013) represent the context of each target word by using the high-probability lexical substitutes according to a statistical language model. A language model was built to identify the relative probabilities of 4-gram sequences and then FastSubs (Yuret (2012)) was applied to identify words that appear in the

same position as the target word for each context. For example, one instance of *bass* may have substitutes relating to the meaning of fish, whereas another instance may have substitutes regarding the instrument meaning of bass. Each instance is then represented as 100 substitutes, sampled from the substitute-word distribution. The instance-substitute pairs are fed into S-CODE (Maron et al. (2010)). The final S-CODE-based vectors are clustered using k -means. Like Schütze (1998), AI-KU requires specifying the number of clusters ahead of time, often setting k to a larger than necessary number. However, to determine the number of senses, AI-KU performs a post-processing step to remove clusters that contain only a few instances, which are likely artifacts of forcing each of the k clusters to be non-empty. The remaining clusters are treated as senses of the word.

Unimelb: Lau et al. (2013) proposed a system that was based on a Hierarchical Dirichlet Process (HDP) (Teh et al. (2006)), a nonparametric extension of Latent Dirichlet allocation (Blei et al. (2003); Steyvers et al. (2004)). HDP automatically infers both the number of topics and each topic’s probability distribution for generating the tokens in the corpus. For sense induction, an HDP model is inferred from contexts of a target word, which produces a distribution over topics for each context. Each topic is treated as a distinct sense of the word. Given a new context of the word, the HDP model can be used to infer its topic distribution, thereby identifying which senses are present. For output, they report the full distribution over senses for each context, weighted by their probabilities.

University Of Sussex (UoS): Hope & Keller (2013b) used a graph-based system for the task. First, using a dependency-parsed version of ukWaC, they constructed a graph where the vertices were k (they found 300 for the best k for this task) of the highest-ranked words that had a dependency relation with the target word. For ranking, they used Pointwise Mutual Information measure (Bouma (2009)). Then, they applied the MaxMax clustering algorithm (Hope & Keller (2013a)) and merged the resulting clusters according to *cohesion* and *separation* (Pang-Ning et al. (2006)) measures.

La Sapienza employed the Personalized Page Rank algorithm [Agirre & Soroa \(2009\)](#) on WordNet-based network and comparing the each sense with the similarity of the context. After that, each sense is ranked by the similarity.

ICE and MSSG: [Kågebäck et al. \(2015\)](#) proposed two systems (ICE-online and ICE-kmeans) based on word embeddings and context embeddings obtained using the Skip-gram Model ([Mikolov, Chen, et al. \(2013\)](#); [Levy & Goldberg \(2014\)](#)), assigning different weights to the context words based on the assumption that context words with higher weights express more about the meaning of a target word. After constructing the embeddings, the k -means clustering algorithm was employed for the former method. In order to choose the k for k -means, they used Pham’s heuristic ([Pham et al. \(2005\)](#)). ICE-online is an extension of the Multi-Sense Skip-gram (MSSG) model. The only difference is in the embedding construction part; MSSG ([Neelakantan et al. \(2015\)](#)) simply uses uniform weights for all context words around the target words, while ICE-online considers the weight and takes into account that some context words have bigger importance than others.

Sense-Topic Model (STM) [Wang et al. \(2015\)](#) proposed sense-topic model for the WSI. They followed the *local approach*, which means each target word was independently treated—sharing no parameters across target words. Their model considers sense and topic as two separate latent variables to be inferred together. Topics are informed by all instances for a particular target word, whereas senses are informed by the local context surrounding the ambiguous word. They used a 21-tokens-length window to distinguish global context words (for topics) and local context words (for senses). They also analyzed the impact of unsupervised data enrichment methods on their sense-topic model.

Table 5.7 shows the performance of each system, with three different evaluation metrics. The AIKU system outperformed other systems according to the supervised evaluation framework. However, STM got the best scores in unsupervised evaluations. Both representations (the bottom two lines in the table) outperformed

System	SE-F1	FBc (%)	FNMI (%)
MFS	57.8	-	-
Random (weighted)	53.3	-	-
AIKU	64.1	39.0	5.32
Unimelb	60.5	47.6	4.82
UoS	60.0	44.7	3.93
La-Sapienza	57.5	-	-
ICE-kmeans	58.6	51.1	5.90
ICE-online	59.1	48.7	5.50
MSSG	59.0	45.9	3.76
STM (ukWaC)	58.4	53.4	7.93
STM (actual)	57.0	57.8	7.27
SCODE _{ESD} (k=5)	61.7	35.8	5.10
Baskaya (k=6)	61.8	33.2	4.75

Table 5.7: A comparison of the accuracies with various systems. SE-F1score stands for the F1-Score of supervised evaluation using Agirre’s method. The top part (rows 1-2) of the table shows the baselines and second group of the table shows the best systems of each participating team in SemEval 2013. The third part (rows 7-11) of the table contains recent systems tested on SemEval 2013 test data. The bottom part of the table contain the performance of our different context representations in local clustering setting.

all participating systems (top 1-4 rows) according to Fuzzy B-cubed. As it can be seen here, there is no consensus between evaluation metrics. For example, AIKU achieves the best supervised score (64.1%); however, its Fuzzy B-cubed score is surprisingly low. Although MSSG gets the lowest Fuzzy FNMI score (3.76%), its score in supervised evaluation is relatively high (59.0%).

Figure 5.3 illustrates the performance analysis of our context embeddings over different cluster sizes by using supervised evaluation framework. Two context embeddings perform closely in local clustering setting, and their accuracies decrease

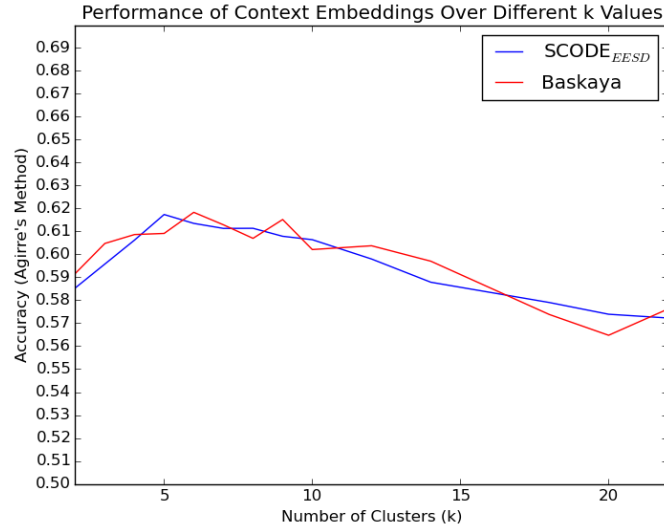


Figure 5.3: Performance analysis of S-CODE context embeddings with respect to the number of clusters in local clustering setting (Agirre’s method is used for evaluation).

while clustering size increases. However, the difference between best and worst score is relatively small ($\sim 3\%$). Figure 5.4 depicts the same analysis with different evaluation metrics – (a) FNMI and (b) FCubed. According to the Figure 5.4, the clustering size highly affects FCubed metric; and it favors to small clustering sizes.

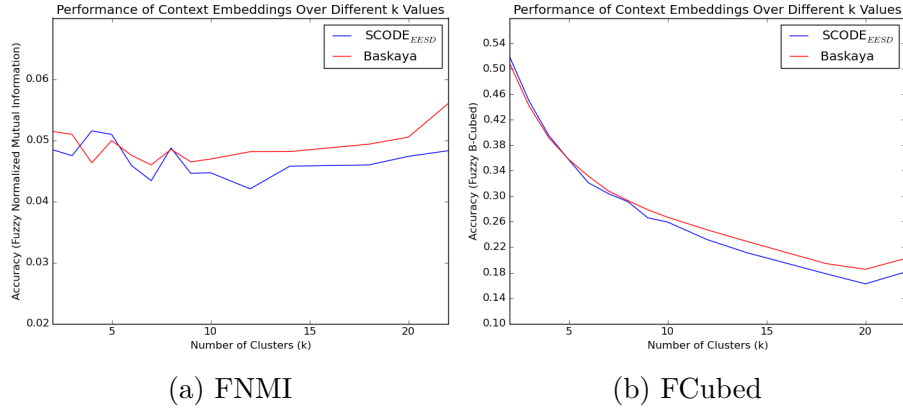


Figure 5.4: Performance differences of S-CODE context embeddings with different number of clusters in local clustering setting. Accuracies are computed by using (a) Fuzzy Normalized Mutual Information and (b) FCubed.

Chapter 6

Conclusion

In this thesis, we applied the similar pipeline described by Yatbaz et al. (2012) which worked well in Unsupervised Part-of-Speech Induction to Word Sense Induction problem. Our system, based on different context embeddings, outperformed participating systems in the SemEval 2010 WSI task, showed competitive performance in SemEval 2013 WSI task, and surpassed the most-frequent-sense baseline in the newly created dataset based on OntoNotes sense annotations. This thesis investigated the performance of different clustering approaches and different context embeddings obtained by S-CODE, as well as popular word embeddings in literature. Using substitute-word distributions, word embeddings in literature can be transformed into context embeddings, which allowed us to compare those word embeddings and S-CODE-based embeddings. Another contribution in this thesis is to provide a Word Sense Induction and Disambiguation dataset using OntoNotes 5.0 annotations. This dataset is highly reliable, and the procedure to create the dataset enables the WSI researcher to compare their results with the proposed methods in this thesis and create their own datasets. This thesis also explored different clustering methods (Spectral Clustering versus DBSCAN versus k -means), clustering approaches (*local approach* versus *pos-based approach*), and different context embeddings (*Baskaya* and $SCODE_{EESD}$) in the context of the Word Sense Induction problem.

This work also showed that evaluation frameworks for WSI systems have discrepancies. It is not unusual that one system seems highly accurate according to one metric, but the same system performs poorly according to another. This problem needs to be addressed in future work, and consensus should be obtained in Word Sense Induction community.

Appendices

Appendix A

Datasets

A.1 SemEval 2010 WSI Task

The test dataset¹ comes from OntoNotes (Hovy et al. (2006)) and consists of various news sources, including CNN and ABC. It contains 100 words (50 nouns and 50 verbs), 5285 noun instances and 3630 verb instances, and 8915 instances in total. The average number of senses for nouns is 4.46 and for verbs is 3.12. Table A.1 summarizes the statistics of the dataset used in this section. The following is an example of the test data instance. The following is an instance from the test dataset:

<swim.v.1>

¹The website of the task contains dataset, scripts for evaluation metrics, and can be found in https://www.cs.york.ac.uk/semeval2010_WSI

	Noun	Verb	All
# of Target Words	50	50	100
# of Instances	5285	3630	8915
Average # of Sense	4.46	3.22	3.84
Average Sense Perplexity	3.00	2.14	2.66

Table A.1: Dataset statistics for the SemEval 2010 WSID task

First of all , visibility will be very very low . <TargetSentence> It 's going to be bitterly cold , and there is going to be enormous danger from jagged pieces of metal which could be swimming around in the submarine . </TargetSentence> Given the fact that there are so many dangers and that these divers are risking their own lives , I wonder if there is consideration given to the fact that this may not be really worth it .
</swim.v.1>

Task organizers also provided a training data set. The training data was used for sense induction, and the final evaluation was applied to the test data. In other words, systems could use only the training data for sense induction.

The evaluation framework consisted of unsupervised and supervised methods. For the unsupervised evaluation, the organizers picked two metrics: *V-Measure* (Rosenberg & Hirschberg (2007)) and *paired-F-Score* (Artiles, Gonzalo, & Sekine (2009)). The organizers provided more details on the choice of these measures in Manandhar et al. (2010). For the supervised evaluation, they used the same evaluation technique with the SemEval-2007 WSI Task (Agirre & Soroa (2007)), except for one difference: the reported results of the SemEval-2010 task were obtained with 5-fold cross validation. These evaluation settings are explained in Chapter 3.

A.2 SemEval 2013 WSI Task

The test set for the *Word Sense Induction for Graded and Non-Graded Senses* task in SemEval-2013¹ was drawn from both spoken and written parts of the Open American National Corpus (Ide & Suderman (2004)) and includes 50 terms (20 verbs, 20 nouns and 10 adjectives) with a variety of genres, such as fiction, non-fiction, technical. There are a total of 4664 test instances provided. All evaluations were performed on test instances only. In addition, the task organizers provided

¹The website of the task, containing the dataset and scripts for evaluation metrics, can be found at <https://www.cs.york.ac.uk/semeval-2013/task13.html>

	Noun	Verb	Adjective	All
# of Target Words	20	20	10	50
# of Instances	1659	1657	806	4122
Average # of Sense	6.5	6.15	4.8	6.02
Average Sense Perplexity	3.37	3.59	3.06	3.40

Table A.2: Dataset statistics for the SemEval 2013 WSID task

sense-annotated trial data which could be used for system tuning. This trial data is a redistribution of the *Graded Sense and Usage* dataset provided by Katrin Erk, Diana McCarthy, and Nicholas Gaylord [Erk & McCarthy \(2009\)](#). It consists of 8 terms (3 verbs, 3 nouns, and 2 adjectives), the number of senses varying between 4 and 7. Each word type in the trial data has 50 contexts. Participants can use ukWaC ([Ferraresi et al. \(2008\)](#)), a 2-billion-word-token web-gathered corpus, for sense induction. Furthermore, unlike in previous WSI tasks, the task organizers allowed participants to use additional contexts from different corpora under the condition that they also submit the same system using only ukWaC, in order to compare participating systems under equal conditions.

The gold standard of test data was prepared using WordNet 3.1 following the work described in [Jurgens \(2013\)](#) with 20 annotators from Amazon Mechanical Turk; however, after realizing the moderate inter-annotator agreement (IAA), the organizers decided to annotate the data by themselves.

Since WSI systems report their annotations in a different sense inventory than WordNet 3.1, a mapping procedure should be used first. The organizers used the sense-mapping procedure explained in [Jurgens \(2012\)](#). This procedure has adopted the supervised evaluation setting ([Agirre, Martínez, ó de Lacalle, & Soroa \(2006\)](#); details can be found in Section 3.2) of past SemEval WSI tasks, but the main difference is that the former takes into account applicability weights for each sense, which is necessary for graded word senses. Although the data contains

graded senses for some instances, I used only the instances that were labeled with only one sense. This exclusion decreased the number of instances used in this thesis to 4122. Table A.2 summarizes statistics of the dataset that was used in this section. The following is an example from the test set. Note that all reported scores through this thesis were obtained using the single-sense instance set.

```
<instance id="add.v.8" lemma="add" partOfSpeech="v" token="added"
tokenEnd="10" tokenStart="5">uh i added up all the taxes that we
were going to pay on all these different specific luxury items and travel
expenses and everything else </instance>
```

A.3 OntoNotes Sense Dataset

WordNet (Miller (1995)) provides a fine-grained sense inventory, which sometimes causes annotations with low IAA since there are cases where the difference between one sense and another is extremely subtle. For instance, Palmer et al. (2007) analyzed the categories of errors humans and WSD systems made by using a fine-grained sense lexicon, and they found that several errors arose because of the subtle sense distinctions but diminished when using the coarse-grained version of the lexicon. Therefore, Hovy et al. (2006) developed a methodology for producing annotations for predicate structure, word sense, ontology linking, and coreference with high inter-annotator agreement (IAA). The sense annotation procedure (see Figure A.1) can be described as follows: First, linguists represent sense distinction using an hierarchical structure like decision tree whose leaves are WordNet senses and parent nodes are more coarse-grained groupings, and they decide whether to merge fine-grained senses into the groupings as one single entity (i.e., a coarse grained group represents all the fine-grained senses below hierarchically of it.). 50 instances of a word are annotated by annotators according to these groupings, and they check the IAA. If IAA is higher than the threshold (90%), then the annotation procedure is done for this batch; otherwise, linguists review the groupings, and the annotation procedure starts from the beginning.

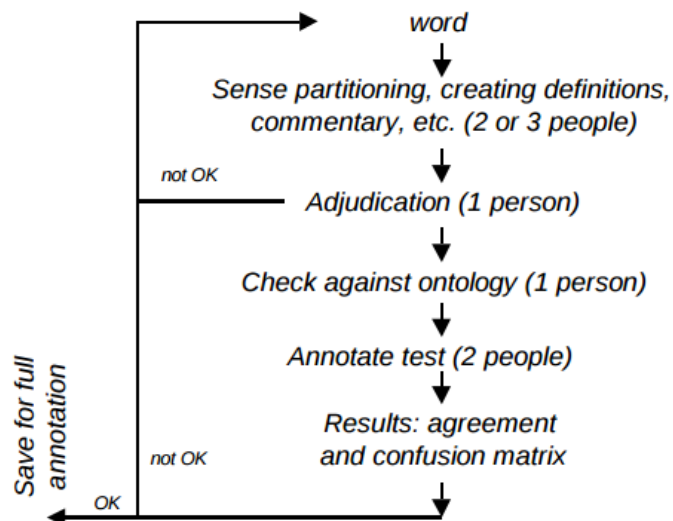


Figure A.1: OntoNotes Sense Annotation Procedure (Hovy et al. (2006))

	Noun	Verb	All
# of Target Words	17	43	60
# of Instances	13544	40209	53753
Average # of Sense	4.82	6.61	6.1
Average Sense Perplexity	2.08	3.27	2.97

Table A.3: Dataset statistics for OntoNotes Dataset

I used OntoNotes 5.0¹. This version covers 1,338 noun and 2,011 verb types. Although they aimed for 90% IAA, some of the word types have lower IAA scores. Therefore, I picked word instances with three constraints: (1) the word has at least 90% IAA, (2) each word type has at least 500 instances, and (3) the sense of the instance is mapped from WordNet 3.0. After applying these constraints, 64 word types remained. I removed word types having only one sense. This decreased the number of word type to 60. Table A.3 summarizes the statistics of the test data.

I also provided a code repository which creates a dataset for Word Sense Induction and Disambiguation problems using OntoNotes 5.0 annotations with different sets of parameters. The parameters are the minimum number of instances for each target word, the minimum IAA level, and one of the word sense inventories used in OntoNotes². Researchers can use the same settings to compare their results with others and this work.

¹The OntoNotes 5.0 Release can be found at <https://catalog.ldc.upenn.edu/LDC2013T19>

²This repository can be found at <https://github.com/osmanbaskaya/induction-disambiguation-dataset-generator>

References

- Agirre, E., Martínez, D., de Lacalle, O., & Soroa, A. (2006). Two graph-based algorithms for state-of-the-art wsd. In *Proceedings of the 2006 conference on empirical methods in natural language processing* (pp. 585–593).
- Agirre, E., Martínez, D., de Lacalle, O., & Soroa, A. (2006). Evaluating and optimizing the parameters of an unsupervised graph-based WSD algorithm. In *Proceedings of textgraphs: the first workshop on graph based methods for natural language processing* (pp. 89–96).
- Agirre, E., & Soroa, A. (2007, June). Semeval-2007 task 02: Evaluating word sense induction and discrimination systems. In *Proceedings of the fourth international workshop on semantic evaluations* (pp. 7–12). ACL.
- Agirre, E., & Soroa, A. (2009). Personalizing pagerank for word sense disambiguation. In *Proceedings of the 12th conference of the european chapter of the association for computational linguistics* (pp. 33–41).
- Ahn, Y., Bagrow, J., & Lehmann, S. (2010). Link communities reveal multiscale complexity in networks. *Nature*, 466(7307), 761–764.
- Amigó, E., Gonzalo, J., Artiles, J., & Verdejo, F. (2009). A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information Retrieval*, 12(4), 461–486.
- Apidianaki, M., & Van de Cruys, T. (2011). A quantitative evaluation of global word sense induction. In *Computational linguistics and intelligent text processing* (pp. 253–264). Springer.

- Arthur, D., & Vassilvitskii, S. (2007). k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual acm-siam symposium on discrete algorithms* (pp. 1027–1035).
- Artiles, J., Amigó, E., & Gonzalo, J. (2009). The role of named entities in web people search. In *Proceedings of emnlp* (pp. 534–542).
- Artiles, J., Gonzalo, J., & Sekine, S. (2009). Weps 2 evaluation campaign: overview of the web people search clustering task. In *2nd web people search evaluation workshop (weps 2009), 18th www conference*.
- Bagga, A., & Baldwin, B. (1998). Algorithms for scoring coreference chains. In *Proceedings of the linguistic coreference workshop at lreca 98* (pp. 563–566).
- Bansal, M., Gimpel, K., & Livescu, K. (2014). Tailoring continuous word representations for dependency parsing. In *Proceedings of the annual meeting of the association for computational linguistics*.
- Baskaya, O., Sert, E., Cirik, V., & Yuret, D. (2013). Ai-ku: Using substitute vectors and co-occurrence modeling for word sense induction and disambiguation. In *Proceedings of the second joint conference on lexical and computational semantics (*sem), volume 2: Seventh international workshop on semantic evaluation (semeval 2013)* (pp. 300–306).
- Biemann, C. (2006). Chinese whispers: an efficient graph clustering algorithm and its application to natural language processing problems. In *Proceedings of the first workshop on graph based methods for natural language processing* (pp. 73–80).
- Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3, 993–1022.
- Bouma, G. (2009). Normalized (pointwise) mutual information in collocation extraction. *Proceedings of GSCL*, 31–40.

- Brody, S., & Lapata, M. (2009). Bayesian word sense induction. In *Proceedings of the 12th conference of the european chapter of the association for computational linguistics* (pp. 103–111).
- Chandler, D. (2007). *Semiotics: the basics*. Routledge.
- Chapman, R. L., Roget, P. M., et al. (1984). *Roget's international thesaurus*. Harper & Row; New York: Harper & Row,[1984], c1977.
- Church, K. W., & Hanks, P. (1990). Word association norms, mutual information, and lexicography. *Computational linguistics*, 16(1), 22–29.
- Cirik, V., & Yuret, D. (2014). Substitute based scode word embeddings in supervised nlp tasks. *arXiv preprint arXiv:1407.6853*.
- Clear, J. H. (1993). The british national corpus. In *The digital word* (pp. 163–187).
- Collobert, R., & Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on machine learning* (pp. 160–167).
- Deerwester, S., Dumais, S. T., Landauer, T. K., Furnas, G. W., & Harshman, R. A. (1990). Indexing by latent semantic analysis. *Journal of the Society for Information Science*, 41(6), 391–407.
- Drucker, H., Wu, D., & Vapnik, V. N. (1999). Support vector machines for spam categorization. *Neural Networks, IEEE Transactions on*, 10(5), 1048–1054.
- Erk, K., & McCarthy, D. (2009). Graded word sense assignment. In *Proceedings of the 2009 conference on empirical methods in natural language processing: Volume 1* (pp. 440–449).
- Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd* (Vol. 96, pp. 226–231).

- Faruqui, M., & Dyer, C. (2014). Improving vector space word representations using multilingual correlation..
- Fellbaum, C. (Ed.). (1998). *Wordnet: An electronic lexical database*. The MIT Press.
- Ferraresi, A., Zanchetta, E., Baroni, M., & Bernardini, S. (2008). Introducing and evaluating ukwac, a very large web-derived corpus of english. In *In proceedings of the 4th web as corpus workshop (wac-4*.
- Firth, J. R. (1957). A synopsis of linguistic theory 1930-1955. In *Studies in linguistic analysis, 1-32*. Oxford: Philological Society. (Reprinted in F. R. Palmer (Ed.), (1968). *Selected papers of J. R. Firth 1952-1959*, London: Longman.)
- Francis, W. N., & Kucera, H. (1979). Brown corpus manual. *Brown University*.
- Gale, W. A., Church, K. W., & Yarowsky, D. (1992). Work on statistical methods for word sense disambiguation. In *AAAI Fall Symposium on Probabilistic Approaches to Natural Language* (pp. 54–60).
- Globerson, A., Chechik, G., Pereira, F., & Tishby, N. (2004). Euclidean embedding of co-occurrence data. In *Advances in neural information processing systems* (pp. 497–504).
- Hope, D., & Keller, B. (2013a). Maxmax: a graph-based soft clustering algorithm applied to word sense induction. In *Computational linguistics and intelligent text processing* (pp. 368–381). Springer.
- Hope, D., & Keller, B. (2013b). Uos: A graph-based system for graded word sense induction. *Atlanta, Georgia, USA*, 689.
- Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., & Weischedel, R. (2006). OntoNotes: the 90% solution. In *Proceedings of the human language technology conference of the naacl, companion volume: Short papers* (pp. 57–60).

- Huang, E. H., Socher, R., Manning, C. D., & Ng, A. Y. (2012). Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th annual meeting of the association for computational linguistics: Long papers-volume 1* (pp. 873–882).
- Ide, N., & Suderman, K. (2004). The american national corpus first release. In *Lrec*.
- Jurgens, D. (2011). Word sense induction by community detection. In *Proceedings of sixth acl workshop on graph-based methods for natural language processing (textgraphs-6)*.
- Jurgens, D. (2012). An Evaluation of Graded Sense Disambiguation using Word Sense Induction. In *Proceedings of *sem, the first joint conference on lexical and computational semantics*. Association for Computational Linguistics.
- Jurgens, D. (2013). Embracing Ambiguity: A Comparison of Annotation Methodologies for Crowdsourcing Word Sense Labels. In *Proceedings of the north american chapter of the association for computational linguistics (naacl)*. ACL.
- Jurgens, D., & Klapaftis, I. (2013). SemEval-2013 Task 13: Word Sense Induction for Graded and Non-Graded Senses. In *Proceedings of the 7th international workshop on semantic evaluation (semeval 2013)*. Association for Computational Linguistics.
- Kågebäck, M., Johansson, F., Johansson, R., & Dubhashi, D. (2015). Neural context embedding for automatic discovery of word senses. In *Proceedings of naacl hlt* (pp. 153–163).
- Klapaftis, I. P., & Manandhar, S. (2008). Word sense induction using graphs of collocations. In *Proceeding of the 2008 conference on ecai 2008: 18th european conference on artificial intelligence* (pp. 298–302).
- Korkontzelos, I., & Manandhar, S. (2010). UoY: Graphs of unambiguous vertices for word sense induction and disambiguation. In *Proceedings of the 5th international workshop on semantic evaluation* (pp. 355–358).

- Kozłowski, M., & Rybiński, H. (2014). Sns: A novel word sense induction method. In *Rough sets and intelligent systems paradigms* (pp. 258–268). Springer.
- Lancichinetti, A., Fortunato, S., & Kertész, J. (2009). Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics*, 11(3), 033015.
- Landauer, T. K., & Dumais, S. T. (1997). A solution to plato’s problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, 104(2), 211.
- Lau, J. H., Cook, P., & Baldwin, T. (2013). unimelb: Topic modelling-based word sense induction. In *Proceedings of the second joint conference on lexical and computational semantics (*sem), volume 2: Seventh international workshop on semantic evaluation (semeval 2013)* (pp. 307–311).
- Levy, O., & Goldberg, Y. (2014). Dependencybased word embeddings. In *Proceedings of the 52nd annual meeting of the association for computational linguistics* (Vol. 2, pp. 302–308).
- Mallery, J. C. (1988). Thinking about foreign policy: Finding an appropriate role for artificially intelligent computers. In *Master’s thesis, mit political science department*.
- Manandhar, S., Klapaftis, I. P., Dligach, D., & Pradhan, S. S. (2010). Semeval-2010 task 14: Word sense induction & disambiguation. In *Proceedings of the 5th international workshop on semantic evaluation* (pp. 63–68).
- Maron, Y., Lamar, M., & Bienenstock, E. (2010). Sphere Embedding: An Application to Part-of-Speech Induction. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, & A. Culotta (Eds.), *Advances in neural information processing systems 23* (pp. 1567–1575).
- Meilă, M. (2003). Comparing clusterings by the variation of information. In *Proceedings of the sixteenth annual conference ofn computational learning theory (colt)*. Springer.

- Mihalcea, R. (1998). Semcor semantically tagged corpus. *Unpublished manuscript*.
- Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., & Khudanpur, S. (2010). Recurrent neural network based language model. In *Interspeech 2010, 11th annual conference of the international speech communication association, makuhari, chiba, japan, september 26-30, 2010* (pp. 1045–1048).
- Mikolov, T., Yih, W.-t., & Zweig, G. (2013). Linguistic regularities in continuous space word representations. In *Hlt-naacl* (pp. 746–751).
- Miller, G. A. (1995). WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11), 39–41.
- Mnih, A., & Hinton, G. E. (2009). A scalable hierarchical distributed language model. In *Advances in neural information processing systems* (pp. 1081–1088).
- Morin, F., & Bengio, Y. (2005). Hierarchical probabilistic neural network language model. In *Proceedings of the international workshop on artificial intelligence and statistics* (pp. 246–252).
- Navigli, R., & Ponzetto, S. P. (2012). Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193, 217–250.
- Neelakantan, A., Shankar, J., Passos, A., & McCallum, A. (2015). Efficient non-parametric estimation of multiple embeddings per word in vector space. *arXiv preprint arXiv:1504.06654*.
- Page, L., Brin, S., Motwani, R., & Winograd, T. (1999). The pagerank citation ranking: bringing order to the web.
- Palmer, M., Dang, H. T., & Fellbaum, C. (2007). Making fine-grained and coarse-grained sense distinctions, both manually and automatically. *Natural Language Engineering*, 13(02), 137–163.

- Pang-Ning, T., Steinbach, M., Kumar, V., et al. (2006). Introduction to data mining. In *Library of congress* (p. 74).
- Pantel, P., & Lin, D. (2002). Discovering word senses from text. In *Proceedings of acm conference on knowledge discovery and data mining (kdd-02)* (pp. 613–619).
- Pedersen, T. (2010, July). Duluth-wsi: Senseclusters applied to the sense induction task of semeval-2. In *Proceedings of the semeval 2010 workshop : the 5th international workshop on semantic evaluations* (pp. 363–366).
- Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. *Proceedings of the Empirical Methods in Natural Language Processing (EMNLP 2014)*, 12, 1532–1543.
- Pham, D. T., Dimov, S. S., & Nguyen, C. (2005). Selection of k in k-means clustering. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 219(1), 103–119.
- Philpot, A., Hovy, E., & Pantel, P. (2005). The omega ontology. In *Proceedings, ijcnlp workshop on ontologies and lexical resources (ontolex-05)*.
- Purandare, A., & Petersen, T. (2004). Discriminating among word meanings by identifying similar contexts. In *Proceedings of the nineteenth national conference on artificial intelligence (aaai-04)* (pp. 964–965).
- Rand, W. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336), 846–850.
- Reisinger, J., & Mooney, R. (2010). A mixture model with sharing for lexical semantics. In *Proceedings of the 2010 conference on empirical methods in natural language processing* (pp. 1173–1182).
- Rosenberg, A., & Hirschberg, J. (2007, June). V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (emnlp-conll)*. ACL.

- Rybinski, H., Kryszkiewicz, M., Protaziuk, G., Kontkiewicz, A., Marcinkowska, K., & Delteil, A. (2008). Discovering word meanings based on frequent termsets. In *Mining complex data* (pp. 82–92). Springer.
- Sahlgren, M. (2006). *The word-space model* (Unpublished doctoral dissertation). Citeseer.
- Salton, G., Wong, A., & Yang, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, 18(11), 613–620.
- Schütze, H. (1998). Automatic word sense discrimination. *Computational Linguistics*, 24(1), 97–123.
- Schütze, H., & Pedersen, J. (1995). Information retrieval based on word senses. In *Proceedings of sdair*.
- Sert, E. R. (2013). Word context and token representations from paradigmatic relations and their application to part-of-speech induction.
- Shi, J., & Malik, J. (2000). Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8), 888–905.
- Sparck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1), 11–21.
- Steyvers, M., Shiffrin, R. M., & Nelson, D. L. (2004). Word association spaces for predicting semantic similarity effects in episodic memory. In W. Kintsch & T. Landauer (Eds.), *Cognitive psychology and its applications: Festschrift in honor of lyle bourne*. American Psychological Association.
- Stolcke, A. (2002). SRILM – An Extensible Language Modeling Toolkit. In *Proceedings international conference on spoken language processing vol. 2* (pp. 901–904).
- Teh, Y., Jordan, M., Beal, M., & Blei, D. (2006). Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101(476), 1566–1581.

- Tibshirani, R., Walther, G., & Hastie, T. (2000). Estimating the number of clusters in a dataset via the gap statistic. *Journal Royal Statistics Society B*, 63, 411–423.
- Turney, P. D., & Pantel, P. (2010). From Frequency to Meaning: Vector Space Models of Semantics. *Journal of Artificial Intelligence Research*, 37, 141–188.
- Van de Cruys, T., & Apidianaki, M. (2011). Latent Semantic Word Sense Induction and Disambiguation. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies (acl/hlt)* (p. 1476–1485).
- van Rijsbergen, C. (1979). *Information retrieval. 1979*. Butterworth.
- Véronis, J. (2004). HyperLex: lexical cartography for information retrieval. *Computer Speech & Language*, 18(3), 223–252.
- Vinh, N. X., Epps, J., & Bailey, J. (2009). Information theoretic measures for clusterings comparison: is a correction for chance necessary? In *Proceedings of the 26th annual international conference on machine learning* (pp. 1073–1080).
- Vlachos, A., Korhonen, A., & Ghahramani, Z. (2009). Unsupervised and constrained dirichlet process mixture models for verb clustering. In *Proceedings of the workshop on geometrical models of natural language semantics* (pp. 74–82).
- Wang, J., Bansal, M., Gimpel, K., Ziebart, B. D., & Clement, T. Y. (2015). A sense-topic model for word sense induction with unsupervised data enrichment. *Transactions of the Association for Computational Linguistics*, 3, 59–71.
- Yao, X., & Van Durme, B. (2011). Nonparametric bayesian word sense induction. In *Proceedings of textgraphs-6: Graph-based methods for natural language processing*.
- Yatbaz, M. A., Sert, E., & Yuret, D. (2012). Learning syntactic categories using paradigmatic representations of word context. In *Proceedings of the 2012 joint*

- conference on empirical methods in natural language processing and computational natural language learning* (pp. 940–951).
- Yatbaz, M. A., Sert, E. R., & Yuret, D. (2014). Unsupervised instance-based part of speech induction using probable substitutes. In *Proceedings of coling*.
- Yuret, D. (2012). Fastsubs: An efficient and exact procedure for finding the most likely lexical substitutes based on an n-gram language model. *Signal Processing Letters, IEEE*, 19(11), 725–728.
- Zaki, M. J., & Hsiao, C.-J. (2002). Charm: An efficient algorithm for closed itemset mining. In *Sdm* (Vol. 2, pp. 457–473).