

**AUTOMATIC WORDNET CONSTRUCTION USING  
WIKIPEDIA DATA**

**VİKİPEDİ VERİLERİNİ KULLANARAK OTOMATİK  
OLARAK WORDNET OLUŞTURMAK**

**FARİD HAZİYEV**

**ASSİST. PROF. DR. GÖNENÇ ERCAN**

**Supervisor**

Submitted to  
Graduate School of Science and Engineering of Hacettepe University  
as a Partial Fulfillment to the Requirements  
for the Award of the Degree of Master of Science  
in Computer Engineering.

2019

This work titled “Automatic WordNet Construction Using Wikipedia Data” by FARİD HAZIYEV has been approved as a thesis for the Degree of Master of Science in Computer Engineering by the Examining Committee Members mentioned below.

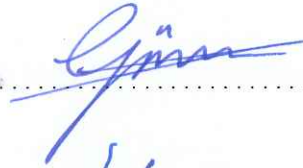
Prof. Dr. İlyas ÇİÇEKLİ

Head



Assist. Prof. Dr. Gönenç ERCAN

Supervisor




Assist. Prof. Dr. Tayfun KÜÇÜKYILMAZ

Member



Assist. Prof. Dr. Mehmet KÖSEOĞLU

Member



Assist. Prof. Dr. Burcu CAN

Member



This thesis has been approved as a thesis for the Degree of Master of Science in Computer Engineering by Board of Directors of the Institute of Graduate School of Science and Engineering on ..... / ..... / .....

Prof. Dr. Menemşe GÜMÜŞDERELİOĞLU

Director of the Institute of  
Graduate School of Science and Engineering


## ETHICS

In this thesis study, prepared in accordance with the spelling rules of Institute of Graduate School of Science and Engineering of Hacettepe University,

I declare that

- all the information and documents have been obtained in the base of the academic rules
- all audio-visual and written information and results have been presented according to the rules of scientific ethics
- in case of using others works, related studies have been cited in accordance with the scientific standards
- all cited studies have been fully referenced
- I did not do any distortion in the data set
- and any part of this thesis has not been presented as another thesis study at this or any other university.

27/06/2019

 Signature  
FARİD HAZİYEV

## YAYINLAMA VE FİKRİ MÜLKİYET HAKLARI BEYANI

Enstitü tarafından onaylanan lisansüstü tezimin/raporumun tamamını veya herhangi bir kısmını, basılı (kağıt) ve elektronik formatta arşivleme ve aşağıda verilen koşullarla kullanıma açma iznini Hacettepe üniversitesine verdiğimi bildiririm. Bu izinle Üniversiteye verilen kullanım hakları dışındaki tüm fikri mülkiyet haklarım bende kalacak, tezimin tamamının ya da bir bölümünün gelecekteki çalışmalarda (makale, kitap, lisans ve patent vb.) kullanım hakları bana ait olacaktır.

Tezin kendi orijinal çalışmam olduğunu, başkalarının haklarını ihlal etmediğimi ve tezimin tek yetkili sahibi olduğumu beyan ve taahhüt ederim. Tezimde yer alan telif hakkı bulunan ve sahiplerinden yazılı izin alınarak kullanması zorunlu metinlerin yazılı izin alarak kullanıldığını ve istenildiğinde suretlerini Üniversiteye teslim etmeyi taahhüt ederim.

Yükseköğretim Kurulu tarafından yayınlanan *Lisansüstü Tezlerin Elektronik Ortamda Toplanması, Düzenlenmesi ve Erişime Açılmasına İlişkin Yönerge* kapsamında tezim aşağıda belirtilen koşullar haricince YÖK Ulusal Tez Merkezi / H. Ü. Kütüphaneleri Açık Erişim Sisteminde erişime açılır.

- Enstitü / Fakülte yönetim kurulu kararı ile tezimin erişime açılması mezuniyet tarihimden itibaren 2 yıl ertelenmiştir.
- Enstitü / Fakülte yönetim kurulu gerekçeli kararı ile tezimin erişime açılması mezuniyet tarihimden itibaren .... ay ertelenmiştir.
- Tezim ile ilgili gizlilik kararı verilmiştir.

27/06/2019

 (imza)  
FARİD HAZİYEV

## ÖZET

# VİKİPEDİ VERİLERİNİ KULLANARAK OTOMATİK OLARAK WORDNET OLUŞTURMAK

**Farid HAZİYEV**

**Yüksek Lisans, Bilgisayar Mühendisliği**

**Tez Danışmanı: Yrd. Doç. Dr. Gönenç ERCAN**

**Haziran 2019, 76 sayfa**

Karşılaştırılabilir yapılar kullanarak WordNet oluşturmak yaygın olarak araştırılmaktadır, ancak Vikipedi'yi bu amaçla kullanmak çok fazla araştırılmamaktadır. Vikipedi, birçok dil için karşılaştırılabilir bir yapıya sahiptir. Bu nedenle bu yapıyı kullanarak, yöntemlerimizi zengin kaynaklı dillere uygulayıp, daha sonra diğer dillerle eşleştirebiliriz.

Bu projede, bir iki dilli ve iki çok dilli yöntem sunuyoruz. İki dilli yöntemimizde Vikipedi'nin yapısı hem doğru synset'leri bulmak hem de onları hedef dile eşlemek için kullanılır. Çok dilli yöntemlerimizde her Vikipedi sayfasında geçen doğru synset'leri bulup ve daha sonra vektörizasyon kullanarak bu synset'leri hedef dildeki kelimelerle eşleştiriyoruz. Çok dilli yöntemlerimizde, WordNet'i olan 14 dili sayfa adlarına göre gruplandırdık ve birkaç çeviriden oluşan Vikipedi sayfalarını oluşturduk. Vikipedi sayfalarında doğru synset'leri bulmak için kural tabanlı ve grafik tabanlı yöntemler kullandık. Vikipedi sayfalarında doğru synset'leri bulduktan sonra, vektörizasyon kullanarak hedef dildeki kelimelerle eşleştirdik. Daha sonra Almanca ve Rusça zemin gerçeği datalarını kullanarak kendi yöntemlerimizi bir biri ile ve

başka state-of-the-art yöntemlerle karşılaştırdık. Sonuç olarak gördük ki bizim yöntemler state-of-art yöntemlere benzer sonuçlar veriyor. Ayrıca daha karmaşık Belirsizlik Giderme yöntemi denendiği zaman sonuçların iyileştiğini gördük.

**Anahtar Kelimeler:** WordNet, Belirsizlik Giderme, Kelime gösterilimi, Vikipedi

## **ABSTRACT**

# **AUTOMATIC WORDNET CONSTRUCTION USING WIKIPEDIA DATA**

**Farid HAZIYEV**

**Master of Science, Computer Engineering Department**

**Supervisor: Asst. Prof. Dr. Gönenç ERCAN**

**June 2019, 76 pages**

Building WordNets from comparable corpora is a task that is explored, but especially using Wikipedia for this purpose is not explored in depth. Wikipedia, has a structure that makes it a comparable corpora for lots of languages. That is why using this structure, we can apply our methods to resource rich languages and then map the results to the resource poor languages. In this paper, we present one bilingual and two multilingual methods. In our bilingual method Wikipedia's structure is used for both finding correct synsets and mapping them to the target language. In our multilingual methods we find correct synsets passing in each Wikipage and then map those synsets to the words in the target language using vectorization. We have grouped 14 languages that have WordNet available for the page names and created Wikipages, where each Wikipage consists of several translations. In order to find the correct synsets in the Wikipages, we used a rule based and a graph based method. After finding correct synsets in each Wikipage, we applied vectorization and mapped those synsets to the words in the translation of the target language Wikipedia. Then we compared

our methods with each other and with some state of art methods using German and Russian languages as ground truth. It is seen that our methods show comparable results to the state of art methods. Also, it is shown that when more complex WSD method is used, our results improved.

**Keywords:** WordNet, Word Sense Disambiguation, Word Embeddings, Wikipedia



# CONTENTS

ÖZET .....	i
ABSTRACT .....	iii
CONTENTS .....	v
TABLES .....	vii
FIGURES .....	viii
SYMBOLS AND ABBREVIATIONS .....	ix
1. INTRODUCTION.....	1
1.1. Overview .....	1
1.2. Thesis Goals.....	2
1.3. Thesis Outline .....	3
2. Background .....	4
2.1. Wikipedia .....	4
2.2. Word Embeddings.....	5
2.3. Word Sense Disambiguation .....	7
2.3.1. Supervised WSD Methods .....	9
2.3.2. Knowledge-based WSD methods.....	10
2.3.3. Selectional Preferences.....	10
2.3.4. Graph Measures .....	12
2.3.5. Semi supervised Methods .....	14
3. RELATED WORK .....	15
3.1. Methods using multilinguality .....	15
3.2. Methods using word embeddings .....	16
3.3. Methods Using Parallel corpora.....	17
3.4. Knowledge Based WSD Methods.....	18
3.5. Supervised WSD Methods .....	23
4. MODEL.....	27
4.1. Bilingual Method - Tagging Page Names, Links and Categories .....	27
4.1.1. Tag page name step.....	27

4.1.2.	Tag page name words .....	30
4.1.3.	Tag categories and links .....	30
4.2.	Multilingual Methods.....	31
4.3.	Rule based Method.....	34
4.4.	Graph Methods.....	36
4.4.1.	Monolingual Graph Method .....	36
4.4.2.	Multilingual Graph Method .....	39
4.5.	Mapping target language to correct synsets using Vectorization.....	42
5.	EXPERIMENTS AND RESULTS .....	46
5.1.	Experimental Setting.....	46
5.2.	Evaluation Metrics .....	46
5.3.	Baseline Methods.....	47
5.4.	Applied Methods.....	48
5.5.	Results.....	48
5.5.1.	Bilingual Method Evaluation .....	48
5.5.2.	Multilingual Rule Based Methods Evaluation .....	49
5.5.3.	Multilingual Graph Based Methods Evaluation.....	50
5.5.4.	Comparison With Baseline Methods .....	51
5.6.	Error analysis .....	52
6.	CONCLUSION.....	54
6.1.	Conclusion .....	54
6.2.	Future Research Directions.....	54
	REFERENCES .....	55
	THESIS ORIGINALITY REPORT .....	61
	CV.....	62

## TABLES

Table 2.1.	Representation of window of words .....	10
Table 4.1.	Example Synset Disambiguation contexts .....	29
Table 4.2.	Toy example .....	36
Table 4.3.	Multilingual Graph Method output .....	43
Table 4.4.	Toy Wikipage Synset Matrix .....	43
Table 4.5.	Toy Word Wikipage Matrix .....	44
Table 4.6.	Tf-idf applied Toy Word Wikipage Matrix .....	44
Table 4.7.	Toy Word Synset Matrix.....	44
Table 4.8.	Sorted example word synset matrix row .....	44
Table 4.9.	k highest scoring synsets for word “demir” .....	45
Table 5.1.	Bilingual Method using different Mapping Methods .....	48
Table 5.2.	Simple Monosemous Method .....	49
Table 5.3.	All Languages Method .....	49
Table 5.4.	Comparison between our Rule Based Methods k = 50 .....	49
Table 5.5.	Monolingual Graph method .....	50
Table 5.6.	Multilingual Graph method .....	50
Table 5.7.	Comparison between Graph Methods k = 50.....	50
Table 5.8.	Comparison between our Methods k=50 .....	51
Table 5.9.	Comparison for different methods .....	51
Table 5.10.	Comparisons to State-of-the-art in Russian Wordnet Construction .....	52
Table 5.11.	Different Wikipedia languages .....	53

## FIGURES

Figure 2.1.	Page name identifier .....	4
Figure 2.2.	Example Disambiguation Page.....	5
Figure 2.3.	Skip-gram Method .....	7
Figure 2.4.	Continuous Bag Of Words Method .....	8
Figure 3.1.	Network structure of method of Yuan et al. ....	25
Figure 4.1.	Page disambiguation context for page Toy .....	28
Figure 4.2.	Multilingual Methods Flow .....	31
Figure 4.3.	Comparable corpora between Danish and English .....	32
Figure 4.4.	Example Wikipedia page to show interlingual links .....	33
Figure 4.5.	Example Wikipage with translations to three languages .....	34
Figure 4.6.	Monolingual Graph For an example page .....	37
Figure 4.7.	Synset Counts after applying the graph method to the first translation ...	39
Figure 4.8.	Synset Counts After applying graph algorithm to all the translations.....	40
Figure 4.9.	Multilingual Graph Method input words .....	41
Figure 4.10.	Multilingual Graph Structure For the toy example .....	42

## **SYMBOLS AND ABBREVIATIONS**

### **Abbreviations**

WSD	Word Sense Disambiguation
IR	Information Retrieval
NLP	Natural Language Processing
POS	Part Of Speech
NER	Named Entity Recognition
LKB	Lexical Knowledge Bases
BFS	Breadth First Search
LSM	Latent Semantic Analysis
SVD	Singular Value Decomposition
CBOW	Continous Bag-of-words
UWN	Universal WordNet
PPR	Personalized PageRank
LSTM	Long short-term memory

# 1. INTRODUCTION

## 1.1. Overview

WordNet was developed as a tool that could be more than a dictionary and will group the words lexically. A group of linguists and psycho-linguists led by G.A. Miller at Princeton University's Cognitive Science Laboratory has built WordNet between 1985 to 1993. Then, it is used for different tasks such as machine translation and information retrieval.

It is a lexical database that groups words in terms of their meanings. These meanings are called synsets. For each synset, its definition, some examples showing its usage in a sentence and its relations with other synsets are available. Relations include hypernymy/hyponymy (super-subordinate relation) and meronymy/holonymy (part-whole relation). Example to super-subordinate relation could be "dog" and "animal" words. Here "dog" is the hyponym of "animal" and "animal" is the hypernym of "dog". Example to part-whole relation could be words "car" and "tire". Here "tire" is the meronym of "car" and "car" is the holonym of "tire".

There are WordNets available other than the Princeton WordNet [1]. For example, EuroWordnet [2] that is built for European languages, such as French, English and Dutch. There is also Balkanet [3], which is built for the Balkan languages, such as Turkish, Romanian, and Greek. Although, there are WordNets available for some languages, for most languages there is no WordNet available, which creates a need for WordNet development. There are two choices, first manually creating WordNets and second automatically developing WordNets. First method is not feasible, as it needs a lot of time and professional linguists and also each language requires this laborious effort. Second method is more suitable, as it does not need as much people to work on and most of the time this method could be applied to all the languages.

There are also two categories for automatically building WordNets. First one is the merge method and second one is the expansion method [4]. In the first method, WordNet is created for the target language from scratch and then mapped to the Princeton WordNet [1]. In the second method, Princeton WordNet [1] is taken as the core WordNet and mapped to the target language via machine translation or bilingual dictionaries. When compared merge method is a more challenging task, that is why expansion method is preferred most of the

time. In this work a novel expand method is proposed. The Princeton WordNet [1] synsets are translated automatically to build WordNets for any language that has Wikipedia and a bilingual dictionary.

In this thesis we try to build a WordNet for any language that has Wikipedia and a bilingual dictionary translating words to English is available. We try to use Wikipedia as a comparable corpora and achieve comparable results with the state-of-the-art methods that use multilingual resources. At first we use the structure of Wikipedia. With that method we only tag page names, links and categories. However, the unstructured text context is a more broad resource and can be used to build WordNets with a higher coverage. For that reason in the next step rule based and graph based Word sense disambiguation (WSD) methods are explored to tag the textual contents of the wiki pages.

## **1.2. Thesis Goals**

Our work can be divided to two parts:

1. Find correct senses in each Wikipage for the source language(s).
2. Map those senses to the words in the target language.

For the first part, different WSD approaches are applied and compared with each other. For the second part a vector space model based similarity method is used as a signal to map the disambiguated synsets to the target language.

Through these experiments, we have investigated the following research questions:

1. Can multilingual resources and source languages improve the results of a bilingual approach?
2. What is the role of WSD in this context and can we improve WordNet construction accuracy by using more complex WSD methods.

### **1.3. Thesis Outline**

In Chapter 2, we give the background information about the methods applied in this thesis. Tools used in this method and knowledge resources are explained.

In Chapter 3, we give basic information about the methods used similar to our work. Different multilingual and bilingual WordNet development methods are explained. Then some WSD approaches are explained. Some of them are similarity based and some of them are graph based. Supervised WSD methods are also explained.

Chapter 4 defines the proposed methods and explain them in depth. Our experiments using Russian and German ground truth WordNets are presented in Chapter 5. Through the experiments a comparison between the proposed methods and the state of the art methods is presented.



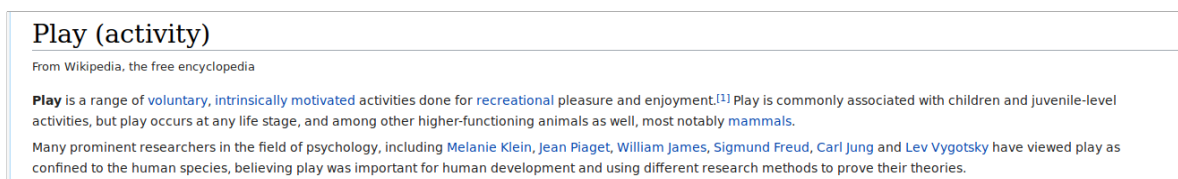
## 2. Background

This chapter introduces important background information that will build the foundations of our proposed methods and the experiments conducted in this work.

### 2.1. Wikipedia

Wikipedia is a free, independent Internet encyclopedia, that is written in many languages jointly by users. It is assumed that it will continue to grow with continuous additions and changes. Since every human being is free to contribute, Wikipedia is getting better in terms of both quantity and quality. Quantity grows as new encyclopedic knowledge is expanded and added to Wikipedia. As each article is seen by huge number of users, mistakes are easily detected and corrected, improving the quality of the articles.

Currently there are 304 languages into which Wikipedia is translated, but 10 of them are closed. Currently 15 languages has more than 1 million articles. Four languages has more than 500,000 articles. Articles are the main concepts that Wikipedia is structured around. Each article in Wikipedia is unique, and if an article name has different meanings, they are disambiguated with identifiers, which are the definitions in bracket near the name of the article. For example, given an article for “play” with the meaning of “a theatrical performance of a drama”, it has a unique identifier play(theatre) which makes it unique (it can be seen in Figure 2.1.). There are also disambiguation pages, where different meanings of a concept are put, such as “play(theatre)” and “play(activity)” (it can be seen in Figure 2.2.).



**Figure 2.1.** Page name identifier

Wikipedia articles have a linked structure. If a word occurs in the article and there is a relevant article about that word, then there is a link from the article to that external page. Sometimes there is also link to dictionary meanings of the word or the translation of the word in another language. There are also some articles which are redirections to other articles.

# Play

From Wikipedia, the free encyclopedia

**Play** most commonly refers to:

- [Play \(activity\)](#), an activity done for enjoyment by animals, including humans
- [Play \(theatre\)](#), structured literary form of theatre

**Play** may refer also to:

---

## Computers and technology [ edit ]

- [Google Play](#), a digital content service offered by Google
- [Play Framework](#), in computer science, an open source web application framework in Java
- [Play Mobile](#), a Polish internet provider and mobile operator
- [Xperia Play](#), released in 2011, an Android phone capable of running PlayStation games

**Figure 2.2.** Example Disambiguation Page

Besides these links, Wikipedia’s structure allows linking to different translations of the article by the use of interlingual links. In our method we define a name as *Wikipage*, which contains all the translations of the same article. One *Wikipage* is several translations of the same article grouped with the name of the English translation of the article. In the future statements, when *Wikipage* term is used it will mean a structure that encompasses all the translations of the same article.

In addition to the linked structure, articles are grouped using the category structures. Categories are all kinds of groups that semantically group articles. For example given a page, named “Rafael Nadal”, its categories are {1986 births, Living people, Australian Open (tennis) champions, French Open champions, tennis players of Spain etc.}.

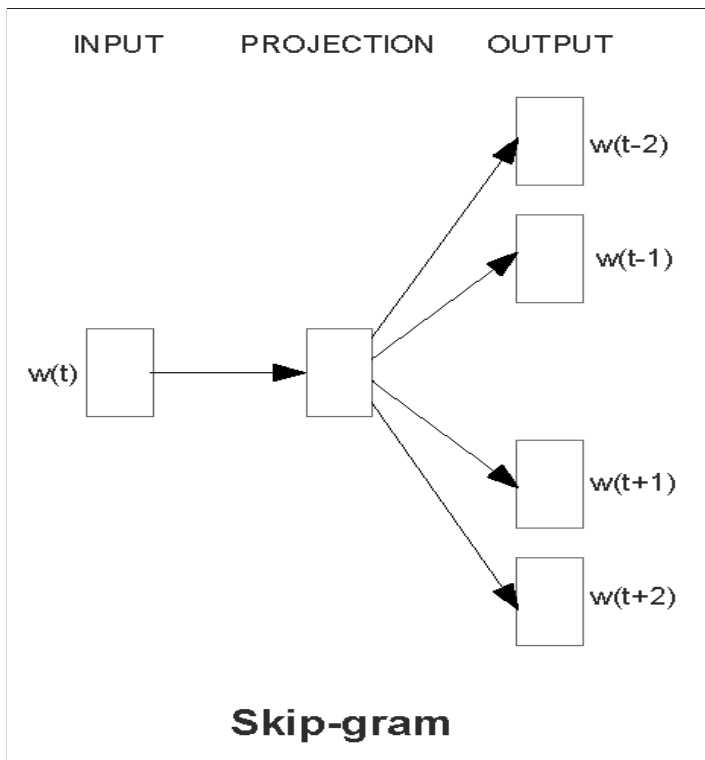
**2.2. Word Embeddings**

According to the Distributional Hypothesis [5], words that have similar meaning occur in the same contexts. In order to find the similarity of two words, similarity of the documents they occur together should be found. Starting in 1960’s Vector Space Models are used for this purpose. Term document matrix is used in this model, which describes the frequencies of words occurring on documents. Term document matrix is a big sparse matrix, which makes it both inefficient and ineffective due to the so called curse of high dimensionality. Also it is usually noisy with rare co-occurrences that only pertain to the local context of the

article. In order to solve these problems, the need to create low dimensional dense matrices emerged. Singular value decomposition based method called as Latent Semantic Analysis [6] is one method that was applied for this purpose. Nearly same time, neural network based methods were also applied. However, only in 2013 with the seminal work of Mikolov [7], word embeddings commonly referred to as word2vec became a common component for most state-of-the-art methods in different NLP tasks. Following word2vec, other word embedding methods like GloVe [8] and fastText [9] have emerged.

To create word2vec, 2 different model architectures are used. These architectures are continuous bag-of-words (CBOW) and continuous skip-gram. In the continuous bag-of-words model, surrounding context words are used to predict the current word. In the continuous skip-gram architecture, current word is used to predict the surrounding words.

We will first explain skip-gram method. The method works as in the Figure 2.3.. A word is fed into a shallow neural network. At first, the word is changed to one hot encoded representation, where it is represented as a vector of size  $|V|$ , which is the vocabulary that consists of all the words in the corpus. In this vector, the index corresponding to the current word  $w(t)$  is set to 1 and all other indices are set to 0. This vector is fed into the neural network. In the first layer of neural network a linear activation function is applied. The number of nodes to chose in the hidden layer is a tunable parameter, but it is chosen as 300 in the original paper. [7]. Another parameter to chose is the context(c), that is to be observed, mostly a sliding window of words near the current word. In the Figure 2.3. it is chosen as 2. In the output layer, a softmax activation function is applied to find the probabilities for each of the words in the vocabulary. The word  $w(t)$  is fed into the model for each word in the context. Using the softmax activation a probability is found for each word in  $V$  coming in the given context location.



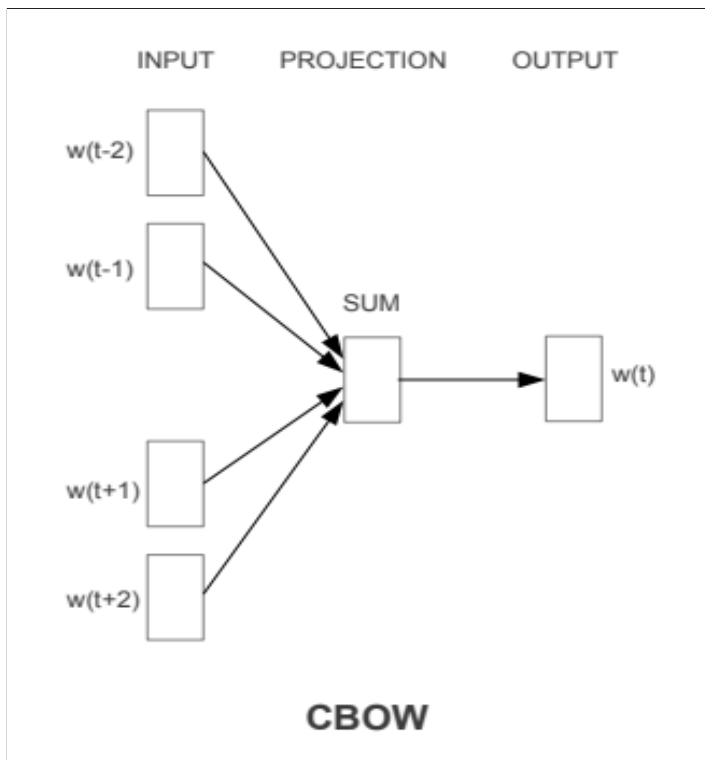
**Figure 2.3.** Skip-gram Method

In the continuous bag-of-words method a similar approach is applied but the architecture of the model is changed like in the Figure 2.4.. Here instead of feeding a vector of size  $|V|$ , a vector of size  $4 * |V|$  will be fed into the model.

### 2.3. Word Sense Disambiguation

Word Sense Disambiguation is a problem, where in the given context, correct sense of the word is predicted from its all possible senses. Its possible senses are taken from a knowledge base, and the most comprehensive one is WordNet. For example, we want to find the correct sense of the word “space” in the given sentence, “Juri Gagarin was the first man to land in space.”. In WordNet, this word has 10 different meanings, but looking at the context, we can say that correct sense is the sense with meaning “any location outside the Earth’s atmosphere”.

WSD can be defined as a classification task, where the word is mapped to a meaning that has highest similarity to it, compared to other possible meanings. However, its structure



**Figure 2.4.** Continuous Bag Of Words Method

differs from other NLP classification tasks, such as part of speech (POS) tagging and named entity recognition. In those applications, there are fixed amount of classes to consider, but in WSD each word has different sets of classes corresponding to their meanings defined in WordNet. For example, for POS tagging the number of classes equals to the number of POS tags. However, each word has different possible synsets, that is why no fixed number of classes could be selected for WSD. For example, if we want to classify a word “word”, it has 11 different senses available in Princeton WordNet [1]. So, looking this as a classification problem, it has 11 classes that is an independent set when considering a word like “space”. To train those classes, we should have a reasonable number of sentences annotated for each sense of that word. So, just this example shows how complex is WSD and its effectiveness highly depends on the available annotated data.

WSD can be viewed from two different perspectives, first trying to disambiguate only one word in the context, which is called as targeted WSD, and second trying to disambiguate all the words in the context, which is referred to as all words WSD. First method is more suitable for supervised methods. In this method, target words are selected and all possible senses are collected from a resource like WordNet. Then those senses are selected as classes and a

classifier is trained for the targeted word using the other disambiguated senses as features. Second method is generally suitable for knowledge based methods, as it needs more data. It is more challenging to model it as a supervised learning task, as the sense of all words contain ambiguity. All possible senses of all words form both the feature space and the solution space.

WSD approaches are also categorized as token-based and type-based. Token-based approach tags all occurrences of the word depending on the context they are used in. Type-based approach tags all the occurrences of the word in the same text with the same sense. Type-based approach can also be considered as an assumption, where a word is used to refer to a single sense in a given context.

WSD depends heavily on the knowledge bases. That is why, several different knowledge bases are used for WSD, which can be divided into two groups, structured and unstructured resources. Examples to structured resources are Thesaurus, Machine Readable Dictionaries, Ontologies. Unstructured resources include corpora, like Brown corpus [10], which is a POS tagged corpus and Wall Street Journal Corpus [11].

In this part, different methods for applying WSD are investigated in depth and some of them are used in our methods. At first, supervised WSD methods are briefly explained, then knowledge based and unsupervised methods are described.

### **2.3.1. Supervised WSD Methods**

Supervised methods learn a model from annotated examples that are constructed manually. Thus, they require a manual effort to annotate a given text corpus with the intended senses. This annotation task is challenging as it both requires the human annotator to understand the general meaning of the text and as it is hard to encode the labeled senses. While for some languages like English a limited corpora exists, resource poor languages lacks these resources and a method targeting to use WSD for building WordNets should first build an annotated dataset.

Supervised WSD methods differ from other classification problems, because in WSD a different training and testing process should be done for each word. In general a classification model has a labeled training data in order to train the model and then predict the labels of the new data that does not have labels. Here labels are the senses of a word. So, for each sense

of a word we should have reasonable number of training sentences. After finding enough data for each sense, the next step is to generate features from those sentences to feed to the classification method. Some examples of the features used in this problem are, vector representation of window of words near the word, part of speeches of the words in the context.

An example to the window approach:

“We went to **play** which was too long.”

Lets suppose, here a window of length 3 is selected. Then the features using POS tags and words could be like this:

The possible representation for this sentence would be table 2.1..

**Table 2.1.** Representation of window of words

$w_{i-3}$	$pos_{i-3}$	$w_{i-2}$	$pos_{i-2}$	$w_{i-1}$	$pos_{i-1}$	$w_{i+1}$	$pos_{i+1}$	$w_{i+2}$	$pos_{i+2}$	$w_{i+3}$	$pos_{i+3}$
went	VBD	to	TO	a	DT	which	WDT	was	VBD	too	RB

### 2.3.2. Knowledge-based WSD methods

Knowledge based methods are the methods, where knowledge bases, such as thesauri, dictionaries, semantic networks (WordNet) are used. From dictionaries, we can get the definition of the word, meanings of the word and some examples for the usages of that meanings. From thesaurus, we can get the synonymy informations of the words. From semantic networks, we can get synonyms, definition, example usage, is-A relations such as hypernymy and hyponymy and part-of relations meronymy, holonymy. Also, sisterhood can be used, which are the synsets that have the same hypernym as the given synset. In knowledge based methods no manually annotated data are used. These methods have bigger coverage as they use relatively larger corpora and try to annotate all open class words in the content.

### 2.3.3. Selectional Preferences

Selectional Preference constrains the way sentences are created. For example after the verb drink we do not expect a solid matter or something that is not a liquid and edible. We expect beverage or alcohol, but not meat or bread. Instead of defining these constraints for each

word, these rules are defined for each semantic class. Semantic classes group the the noun synsets. Nouns are divided to two groups, concrete nouns and abstract nouns. Concrete nouns are also divided to four groups, proper names, common nouns, material nouns and collective nouns. Selectional preferences define the probability of using a class with a given verb.

Philip Resnik [12] used Selectional Preferences for Word sense disambiguation. He used probabilistic approach to show the co-occurrence behaviour of predicate and the semantic classes. For example, a word "person" has a higher probability than "insect" in a verb-subject relationship. However if the predicate is "buzz", then posterior probability of "insect" will increase and "person" will decrease. The difference between posterior and prior probabilities define the effectiveness of the selectional preferences. The strength of predicate in selectional preferences is calculated as:

$$S_R(p) = \sum_c P(c|p) * \log \frac{P(c|p)}{P(p)} \quad (1)$$

(c: class, p: predicate)

The contribution of a semantic class to a predicate's selectional effectiveness is calculated as:

$$A_R(p, c) = \frac{1}{S_R(p)} * P_R(c|p) * \log \frac{P_R(c|p)}{P(p)} \quad (2)$$

$$P_R(c|p) = \frac{\text{count}_R(c, p)}{\text{count}(p)} \quad (3)$$

$$\text{count}_R(c, p) = \sum_{w \in c} \frac{\text{count}_R(p, w)}{\text{classes}(w)} \quad (4)$$

$\text{classes}(w)$  : number of classes  $w$  belongs to.



$count_R(p, w)$  : number of times  $w$  and predicate is in predicate-argument relationship.

They use this idea in order to chose which synset to map to the word. Given a word  $n$ , that has  $PS = [s_1, s_2, \dots, s_k]$  as its possible synsets. They calculate the score of  $sense(i)$  to predicate as:

$$a_i = \max_{c \in C_i} A_R(p, c) \quad (5)$$

where  $C_i$  is all classes of the sentence  $s_i$ .

Then their algorithm works as selecting the sense if it is the only sense for the word, else selecting the sense that has the highest score.

#### 2.3.4. Graph Measures

A popular WSD approach is using Graph theory. In this approach the most important node for each word is found, where nodes are the senses. Methods that use this approach are knowledge based methods, as they need a lexicon for the possible senses and for finding the relations between those senses. Most common used lexicon is WordNet. Graph methods are different from the similarity based methods, that are explained before. In similarity based methods, we try to find the most appropriate sense for each word separately using the similarity with the context. However, in graph based methods, we try to find appropriate senses for all the words at the same time, by taking other words into account. Graph based methods are superior to other unsupervised methods as it takes a bigger context into account to find the correct sense. These methods are not as accurate as supervised methods. However, in contrast to supervised methods, graph methods does not need a labeled dataset. Also, supervised methods have smaller scope even for resource rich languages.

In order to calculate the scores for the senses, different methods are tried. The scoring methods are categorized as global and local. Local measures shows the connectivity of a vertex to the graph, but global measures shows the total connectivity. Local measures include Degree centrality [13], Eigenvector centrality [14], Key Player problem [15] and Betweenness Centrality [16]. Global measures include Compactness, Entropy and edge density.

Degree centrality measures the importance of a node in terms of its degree in the graph. Degree of a node is the number of edges incident to the node.

$$C_D(node) = \frac{deg(node)}{|V| - 1} \quad (6)$$

Here  $V$  is the degree of the graph and the function  $deg$  returns the degree of the node.

Eigenvector centrality measures the influence of a node in the network. In this measure connecting to high-scoring nodes contributes more to the nodes' score than connections to low-scoring nodes. PageRank [17] and HITS [18] are the methods used for this calculation. PageRank [17] scores each vertex according to its importance in the graph. HITS [18] calculates hub value and authority value for each node. A hub score calculates the importance of the node in terms of the nodes it points to. But authority score, calculates it in terms of the nodes pointed to the node.

PageRank [17] score is calculated as:

$$PR(v) = \frac{1 - d}{|V|} + d * \sum_{u \in M(v)} \frac{PR(u)}{outdegree(u)} \quad (7)$$

The formula can be explained as follows, score of vertex  $v$ ,  $PR(v)$  is calculated. The parameter  $d$  is the damping factor and  $|V|$  is the number of vertices in the graph.  $M(v)$  is the set that represents the vertices connected to vertex  $v$  and  $outdegree(u)$  is the number of outgoing links from vertex  $u$ .

Key player problem method gives more importance to nodes that are close to all other nodes in the graph.

$$KPP(v) = \frac{\sum_{u \in V: u \neq v} \frac{1}{d(u,v)}}{|V| - 1} \quad (8)$$

$d(u,v)$  is the shortest path from node  $v$  to node  $u$ . If a node is a disconnected node, its score is calculated as  $\frac{1}{|V|}$ .

Compactness is a global measure and calculates the compactness of the graph, meaning how easily each node is reached by other nodes.

$$CO(G) = \frac{Max - \sum_{u \in V} \sum_{v \in V} d(u, v)}{Max - Min} \quad (9)$$

Graph Entropy measures if the vertices are equally important or not.

$$H(G) = - \sum_{v \in V} p(v) \log(p(v)) \quad (10)$$

Edge Density measures the amount of edges in the graph.

$$ED(G) = \frac{|E(G)|}{\binom{|V|}{2}} \quad (11)$$

$\binom{|V|}{2}$  is the maximum possible numbers of edges.

$E(G)$  is the actual number of edges.

### 2.3.5. Semi supervised Methods

In semi supervised methods, the general idea is to use a small annotated data and extend that data using some bootstrapping methods. Co-training and self-training are probably the most common two semi supervised learning approaches. A subset of a large unlabelled dataset is selected randomly and classified using self training or co-training classifiers, trained on a small labeled dataset. After the labels are assigned to the new samples, samples that were classified with a high confidence are selected and added to the training data. These methods have been used in the context of word sense disambiguation.

### 3. RELATED WORK

In this part, different related works on automatically constructing WordNets are investigated. They are grouped as methods that use multilinguality, methods that use word embeddings and methods that use parallel corpora. Also, different WSD methods are investigated and explained in supervised, knowledge-based and unsupervised categories.

#### 3.1. Methods using multilinguality

Lam et al, [19] uses machine translation and publicly available WordNets in their method. They translate synsets of existing WordNets for different languages to the target language. They take lemmas for all the languages with WordNet and translate them to the target language. Translations are extracted in three ways. First, method is direct translation, where translations are extracted directly. Second, using intermediate WordNets. Third translating other WordNet to English, that has translation to the target language and then translate that WordNet to the target language. The possible words for a synset in target language are ranked in terms of a ranking score and highest scoring one is selected.

In their work Sagot and Fiser [20] use multilingual resources together with the Princeton WordNet [1] to build a French WordNet. They applied 2 different approaches, alignment and translation. In alignment approach, they built a broad multilingual lexicon. All possible synsets of the words in different languages are generated and the synset that takes place in all the languages is mapped to the French translation. Then translation approach is applied to all the monosemous words. Using bilingual dictionaries monosemous words are translated to French.

In another method Taghizadeh and Faili [21] build a WordNet for Persian applying cross lingual WSD. They use only a bilingual dictionary and a monolingual corpus. The WSD problem is modeled as a probabilistic model. They take words from a Persian corpora and translate them to English. Then they extract all possible synsets of these words. Then they use an expectation maximization approach to get the probability of each word synset pair and select the synsets above the threshold score.

In the research of Patanakul and Charnyote [22], a semi-automatic expanding approach is presented to construct a Thai WordNet. Links between Thai words and WordNet synsets are

derived from WordNet and its translations. To rank the links, 13 criteria are used. These criteria are categorized into three groups: monosemic, polysemic, and structural criteria. Monosemic criteria includes, Monosemic one-to-one, Monosemic one-to-many, Monosemic many-to-one and Monosemic many-to-many criterions. Polysemous criteria includes, Polysemic one-to-one, Polysemic one-to-many, Polysemic many-to-one and Polysemic many-to-many. Structural criteria includes, Variant, Intersection, Parent, Brother, Distance hyponym criterions.

In their method De Melo and Weikum [23] use previously known WordNets and dictionaries to build a Universal WordNet (UWN). Their method starts by building an initial graph using existing WordNets and dictionaries. In that graph, words and senses are the nodes and edges are their relations, such as translation or WordNet relations. An SVM classifier is trained to map the words to their synsets.

In another method Bond and Foster [24] use open license WordNets and Wiktionary for building a multilingual WordNet. It starts by taking those WordNets and linking them to the Princeton WordNet [1]. Then it is extended by linking Wiktionary senses to PWN synsets.

In another method Ercan and Haziyevev [25] use Wiktionary and available WordNets to build WordNets for any language automatically. They used one greedy unsupervised method and a supervised method. In the greedy method, they build a graph using the Wiktionary translations and the current WordNets. Nodes are the words and edges are the translations. At first clusters are formed from nodes that has synsets available for them. Then looking at their similarities to these clusters new nodes are added to the clusters and they are iteratively expanded to new words. In the supervised method, a binary classification method is trained using the words from available WordNets. The binary classification task is defined for each word-synset pair, and classifies if the word should be a member of the synset or not. 10 different features are used, that are taken from the graph relations.

### **3.2. Methods using word embeddings**

Word embeddings are commonly used to represent the words meanings. However, all senses are encoded with a single vector. They are used to model the semantic information within a context, and used for disambiguating the intended sense of the words in a text.

In their method, Tarouti and Kalita [26], used the method of Lam et al. [19] to generate synsets. However, that method does not generate semantic links between the synsets. This method tries to add the links and remove some words from the synsets that seems to be irrelevant. To do this, they use word embeddings. The method works in two parts. First, they remove irrelevant words from synsets. To do this, they calculate the cosine similarities between the embeddings of the words in the synset. If the highest similarity is below the threshold  $t$ , then that synset is removed, otherwise they look at all the words and if a word's highest similarity is below that threshold then that word is removed from the synset. If a synset does not have any word left, then it is also removed. Second, they look at the similarity between the words of two semantically related synsets. If the highest similarity is below the threshold  $t_p$ , then that relation is removed.

In another method Khodak et al.[27], use a bilingual dictionary and word embeddings. They apply different strategies for calculating similarity scores between words and synsets. First, they use a naive approach and take the average cosine similarity of the target word and all the lemmas of the synset. Second, they improve this by changing synset representation. They use relations, definitions and example sentences of the synset. They calculate a vector from these data, using the sentence embedding formula of Arora et al. [28]. Then they apply a word sense induction method to solve problems caused by polysemous words.

In their method, Sand et al., [29], uses word embeddings to add synsets and hypernymy relations to existing WordNet. Word embeddings are trained on a large news text data. They identify candidate hypernyms by selecting ancestors of their nearest neighbors on WordNet. Then those hypernyms are scored according to distributional similarity and distance in the WordNet graph.

### **3.3. Methods Using Parallel corpora**

There are also methods that use parallel corpora for WordNet construction. One of them is the work of of Oliver and Climent [30], where they use parallel corpora. If no parallel corpora exists for the language pair, exists or they create one using a machine translation system. Their algorithm starts with sense-tagging the English part of the parallel corpora using Freeling and UKB [31]. Then using a POS tagger they tag the target language part of

the parallel corpora. Then, they use an alignment algorithm to align the sense tagged English part to the target language.

Another method using parallel corpora for WordNet development was applied by Saveski and Trajkovski [32]. They have built a Macedonian WordNet, consisting of 17,553 words and 33,276 synsets. As there was no parallel corpora available between Macedonian and English, they used Google's translation tool. Their method should also solve the WSD problem and it is similar to other methods using WSD for WordNet development. They start with Princeton WordNet [1], translate it to Macedonian and apply WSD algorithm to decide which word maps to the synset. For WSD, they used gloss of the synset and translated it to Macedonian. Candidate words are compared with the gloss using Google similarity distance score [33]. Ones higher than the threshold are selected.

Lee et al [34], use bilingual dictionary and Princeton WordNet[1] to build a Korean WordNet. They start with sense of a Korean word and select the WordNet synsets of the translations from English as candidate synsets. To correctly map the synset to the word, they state that one method is not enough and calculate 6 different heuristic scores and then use those scores as features in a Decision tree for WSD.

Mousavi and Faili[35] use a Persian corpus, Persian WordNet (FarsNet), Princeton WordNet [1] and bilingual dictionary. Using a bilingual dictionary, Persian words are translated to English and using the synsets of those words in Princeton WordNet [1], initial links are created. As their corpus is a POS tagged corpus, they were able to remove the links between words and the synset, when the word does not have the given POS tag. Then for all the words, context vectors are created. These contexts are created using 100 words that occur most with the given word. Then using those context vectors 7 different features are calculated. Then those features are used to construct a classification system where FarsNet is used as the training data.

### **3.4. Knowledge Based WSD Methods**

Although, supervised methods give better results compared to knowledge based methods in targeted WSD tasks, most of the time knowledge based methods give better results for all words tasks. Also supervised methods are not applicable to resource poor languages, unlike knowledge based methods. These are reasons that we also used knowledge based

WSD methods. The most used WSD methods are similarity based methods and graph based methods.

One of the most used and well known WSD method is LESK [36] algorithm, which is used with knowledge bases. This method tries to find the correct senses of the words in the context, by looking at the definition overlaps of their senses. All the sense combinations are tried and the one that gives the highest overlap is selected. Applying this method would be very expensive, because all the possible sense combinations should be tried. Lets calculate how many combinations should be tried for a context with 10 words and each has 5 possible senses. 9,765,625 different combinations should be tested. To cope with this, Cowie et al. [37], applied simulated annealing method.

Simulated annealing[38] is a method that tries to minimize the complexity of large scale combinatorial problems. There is an E value, named Energy, which is calculated by the combination of word senses in the context. E value is calculated iteratively. At each iteration a new configuration of word senses is selected randomly and the Energy value is calculated again. If the Energy value is less than the old value, the configuration is replaced with the new one and iteration continued. Even if the Energy is higher than the old value, the new configuration could be chosen. A probability value is calculated to decide to change the configuration or not. This makes the method not to stuck on local minimum. After each iteration the probability decreased making it less possible for worse configuration to be selected. After some iterations, if there were no change in the configuration for a long time, it is stopped and final configuration is selected.

In [37] this is applied as follows:

- All the words in the context are stemmed
- All possible synsets of the words are taken.
- For all the words their most used synsets are taken as the first configuration.
- To calculate Energy score, at first Redundancy score R is calculated by giving a score  $n - 1$  to stemmed word that appears  $n$  times in the definitions of the selected senses, and add them together. Energy score is calculated as the inverse of R score,  $\frac{1}{R+1}$ . So, as the redundancy increases the energy score decreases. This is chosen as the starting Energy and configuration.



- Iteratively, randomly a word and one of its synsets other than the current synset is selected and energy is recalculated.
- If  $\Delta E$ , change in energy with respect to the old value, is negative then configuration is changed. If,  $\Delta E \geq 0$ , then a probability score is calculated, which decides to change the configuration or not. Probability score is calculated as,  $P = e^{\frac{\Delta E}{T}}$ , where T is a constant which is 1 initially. However, after each time  $\Delta E > 0$ , T is replaced with  $0.9T$ . This process continues, until, after  $n \times 1000$  iterations there was no change in the configuration.

One variation of LESK algorithm [36] is Simplified LESK algorithm [39]. The original Lesk algorithm is about finding the correct senses for all the words in the context at the same time, measuring the definition overlap between all of them. It is very complicated, that is why another simplified Lesk Algorithm is also applied. Here, instead of finding correct senses for all the words at the same time, it is calculated for each word separately. Also calculation process is different. Instead of calculating the overlap of all the definitions, for a word its all possible synsets are extracted, and its overlap with the current context is calculated. The synset with the highest score is selected.

Another variation of LESK [36] is Corpus Lesk Algorithm [40]. This method is also similar to simplified Lesk Algorithm [39]. However, here we should have a sense labeled data available. For each sense, take all the sentences that is labeled with that sense and add to the gloss and examples of that sense and call it the signature of the sense. Instead of finding overlap of the context with the gloss, find the overlap of the context with signature.

Mihalcea [41] applied knowledge based method. She used Wikipedia to construct a sense tagged corpus. She used hyperlinks in Wikipedia pages for WSD purposes. She stated that, Wikipedia is like a sense tagged corpora, because of its interlinks. Interlinks help to disambiguate the meaning of the word. At first, she start by extracting the paragraphs from the Wikipedia, that contains ambiguous words. She does not take words with upper case into account, as named entities are not extensively stored in WordNet and are not usually ambiguous. Then, she collects all the possible senses for the ambiguous words by taking the word in the link. Finally, she manually maps the senses to the WordNet senses. As an example, for the ambiguous word “bar”, she has extracted 1217 paragraphs. Then she has removed the paragraphs that contain only the word bar as disambiguation. Labels that are left are manually mapped to 9 possible WordNet senses for that word.

Graph based methods are applied very often for this task. In their graph based method Navigli and Lapata [42], create graphs for all the sentences that they try to tag. In these graphs nodes are the senses and edges are the semantic relations in the lexicon. Generally used lexicon is WordNet, but they also try an extended WordNet that they refer to as EnWordNet [43] and compare their performances. The method works as follows for each sentence:

- Get all the possible senses in the sentence (only open class words are taken into account).
- For each sense make a dfs in the lexicon and whenever a sense in the sentence is met add all the intermediate nodes and edges to the graph.

After the graph is constructed, scores are calculated for each sense using local and global measures and results of these measures are also compared with each other. Sense with the highest score is selected for a word.

Another graph based method is the method of Ponzetto and Navigli [44], where they grow the WordNet at first using Wikipedia and then apply two different WSD algorithms. One method uses simple Lesk algorithm and another uses graph algorithm. In order to extend the WordNet they use semantic relations in Wikipedia. At first, they make a mapping between Wikipedia pages and WordNet senses. Each page name is tagged with a sense. In order to do this, at first a disambiguation context is created for each Wikipedia page and possible senses. Disambiguation context for the page name include, explanations in the title, links to external pages in the content and Categories of the page. Disambiguation context of word  $w$  is given as  $Ctx(w)$  that includes words in those resources. Then, a disambiguation context is created for each sense. Synonyms, Hypernyms/hyponyms, sisterhood and gloss of the sense is added to its disambiguation context. Sisterhood of a sense are senses that have the same hypernym with that sense. Words in the gloss of the sense are also added. For a sense  $s$ ,  $Ctx(s)$  is all the words in its disambiguation context. The mapping algorithm works as follows:

- Articles that are monosemous, meaning have only one sense are mapped to that exact sense.
- Then looked at all unmapped Articles. For each mapped Article that is redirection to the unmapped Article, its sense is mapped to unmapped Article if that sense is in the possible senses of the unmapped Article.

- For the Articles that have not been assigned yet, looked at  $P(s \rightarrow w)$  of each sense for that word and the one with the highest probability is mapped to that word. If there is a tie, then that Article is not mapped. Because  $P(s|w) = \frac{P(s,w)}{P(w)}$ , calculating  $P(w)$  is irrelevant, so finding  $P(s,w)$  is enough.

$$P(s, w) = \frac{score(s, w)}{\sum_{\substack{s' \in Senses_{WN}(w), \\ w' \in Senses_{Wiki}(w)}} score(s', w')} \quad (12)$$

The score is calculated as:

$$score(s, w) = |Ctx(s) \cap Ctx(w)| + 1 \quad (13)$$

After tagging Articles with correct synsets, the next step is to add the relations between the Articles to the WordNet. If two pages have links to each other and page names of both are tagged with WordNet synsets, then their relation is added to WordNet. They call the extended WordNet as WordNet++. Then they use WordNet++ with two different WSD approaches. First they use simplified Lesk algorithm and then a graph algorithm using the degree centrality measure.

Mihalcea [45] uses page rank algorithm in her graph based method, in order to score the synsets. The method starts by taking all the senses of the words in a text. These senses are shown as vertices in the graph. Then edges are added for all the senses of a word with respect to all the senses of the words in a given window in the text. Edges are calculated as weighted edges. To do this intersections of the definitions of the synsets from WordNet are taken. Then, number of intersection is divided by the length of the definitions of both senses. This way a weighted graph is created. In the final step, PageRank algorithm is applied. Words are mapped to their highest scoring synset in the graph.

Agirre et al. [46] uses graph based method with a personalized page rank algorithm. PageRank algorithm has a smoothing factor part  $(1 - d)v$ . Here  $v$  is a vector  $[\frac{1}{n}, \frac{1}{n} \dots \frac{1}{n}]$ , which has a uniform distribution. In random jumps there is equal probability to jump to any node. If the value of any index is increased, then the node corresponding to that index will have a higher chance to be reached in any random jump. This way its importance will increase. This will also increase the importance of other nodes that are highly connected to this node. They used 3 different lexical knowledge bases (LKB), The Multilingual Central Repository, WordNet 1.7 and WordNet 3.0 for comparison. Their algorithm starts by building a large

graph from LKB, where nodes are the synsets in the LKB and edges are all the relations between the synsets. Then given an input text, all the open class words are extracted and all their possible synsets in the lexicon are added to a list. Then two different methods are tried and compared. First one uses PageRank [17] method. In first method, all the senses in the context are searched with BFS(breadth first search) in the LKB and their minimum distances to all other senses in the context are calculated. Then a subgraph is created, where nodes are the senses in the context and edges are the minimum distance scores. Then PageRank [17] algorithm is applied to the created graph. Second, they apply Personalized Page Rank on the whole graph of LKB. They give the direct PageRank [17] equation like this:

$$Pr = cMPr + (1 - c)v \quad (14)$$

Here  $c$  is the damping factor,  $M$  is the transition probability matrix and  $v$  is a matrix that shows the probabilities of the random jump of each node. In the PPR(personalized page rank) giving higher values to some nodes in vector  $v$ , then most of the random jumps will come to those nodes increasing their rank. They start by adding words in the context to the graph and connect them with directed edges to their possible synsets. Then give high initial probabilities to the word nodes and apply PPR. They state that, PPR method has a problem, that when there is relation between possible synsets of a word, it may dampen the effect of other synsets in the context. To cope with this they devise a variant of ppr and apply a different method and call it ppr\_w2w. They give higher initial probabilities to the senses of the words surrounding the target word. Comparing these method ppr\_w2w gives the best results.

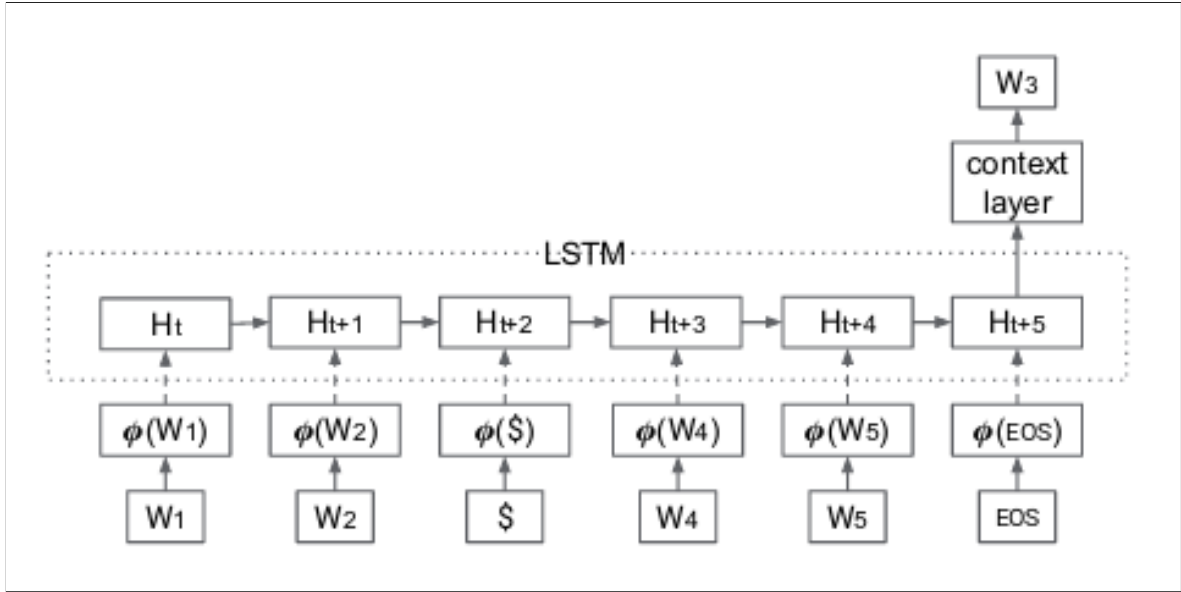
### 3.5. Supervised WSD Methods

Supervised methods applied on WSD gives the best results most of the time. One such method is proposed by Kageback et al. [47]. They state that, they have not used any external knowledge resource, compared to other state of art methods and achieved comparable results. Most of the methods does not take into account the sequence but take the content as a bag of words. They give a good example to show the importance of word sequence on disambiguating the word sense : "Hard rock crushes heavy metal". In this sentence, if the

words are considered separately then it is possible to conclude that the meaning of the rock is a stone. However, looking at the sequence we can say that it is about music. They try to address two problems of previous WSD methods. First one is integrating sequence information instead of using bag of words assumption. Second one is reducing the dependency of the method to external resources such as knowledge bases. They solve first problem by building a sequential model and for the second problem they solely depend on word embeddings instead of a knowledge base. They used bidirectional long short-term memory network which is shared between all words. Bidirectional LSTM is an adaptation of LSTM, where state consists of two LSTMs, one going to the left and one going to the right. This way, the model keeps the information about preceding words as well as the proceeding words. GloVe Word embeddings[8] are used as the input to the bidirectional LSTM network. The model of the LSTM network consists of a softMax layer, hidden layer and a bidirectional LSTM.

Pasini and Navigli [48], also applied supervised method for Word Sense Disambiguation. Their method is called Train\_O\_Matic, which only needs WordNet as a resource. It is superior to other supervised methods, because no manual annotation is needed. Their method is language independent as they use a multilingual resource called as BabelNet [49]. The method consists of three parts: Lexical Profiling, Sentence scoring and Sentence ranking and selection. In the Lexical Profiling part, they compute the relatedness between senses using Personalized Page Rank algorithm. In the end of lexical profiling they have created relatedness probability between each sense of a word. In the sentence scoring part, they try to find the importance of the senses for the given sentences. In the final step sentence selection step, for each sense the sentences where it is the highest scoring sense are selected. Then those sentences are ranked.

Another supervised method applied is the method of Yuan et al., [50], where two methods are used for Word Sense Disambiguation, LSTM method and semi-supervised label propagation. Method using LSTM is shown to outperform other methods that use bag of words approach. In the label propagation model, they tag the words that are not tagged with LSTM. They do not predict the sense directly using the LSTM model, instead they predict the words. They use a very huge training set consisting of 100 billion tokens. A sentence is selected, and the purpose is to predict the held-out word in a sentence. They replace it with \$ and project them to a h dimensional hidden layer and project the hidden layer to p dimensional context layer and finally to the softmax layer that predicts the held-out word. Their network is given in 3.1.



**Figure 3.1.** Network structure of method of Yuan et al.

After training the model, the next step is to predict the senses for each occurrence of a word. Context vectors are created for each context, of the word. They have selected the context layer, as the context vector. To find the sense vectors, average of the context vectors of all the training sentences that are labeled with the given sense is calculated. Then to map the word to the correct sense, cosine similarities of the sense vectors with respect to the context vector is calculated and sense with the highest score is mapped to the word. To overcome the need for a huge number of training data, label propagation is also used. In this method a number of sense labeled sentences are augmented with the unlabeled sentences. Label propagation is built as a graph with some nodes labeled and some not. Then sense labels are propagated to the unlabeled sentences. When LSTM model is used together with the label propagation method, best results are achieved.

Another supervised method is applied by Dandala and Mihalcea [51], where they have applied 3 different supervised approaches using Wikipedia. First they have applied a monolingual method. To create sense annotated training data, they use Wikipedia hyperlinks. They take all the sentences where the ambiguous word is present and using the approach of Dandala et al. [52], build the training data. Then the training data is tokenized and labeled for POS tags. They take the current word and window of three words to the left and right as features for the supervised model. They also add words that occur at least three times in the sentences for the given sense. Then they use naive Bayes classifier to train the model. In their second method which they call WikiTransSense, they translate sentences in the target

language to another language using machine translation and word alignment and train the model using both languages. In their final approach, they use multilinguality of Wikipedia. For example, given a page “bar (music)” in English, there is a translation page “takt (musik)” in German. The senses are used to create sense tagged corpora for German.

Navigli et al. [53] use BabelNet [54] for multilingual word sense disambiguation. In their method, they manually tag the English source using BabelNet [54] synsets. Other language words are tagged using the multilingual links of BabelNet [54]. Aligned sentences between English and non-English languages are selected. For each word in non-English sentence, all of its possible synsets are searched in the synsets of the English tagged sentence and if the synset is found, it is mapped to the word.

## 4. MODEL

WordNet is a well known topic, where both bilingual and multilingual methods are tried. We started with a bilingual method where only page names, links and categories are matched with the correct synsets. In our bilingual method, Wikipedia is used as a comparable corpora and words whose synsets are found in the English translation are directly mapped to their translation in the target language using bilingual dictionary.

Then we have applied 2 different multilingual methods. In those methods, multilinguality of Wikipedia is used. Instead of directly finding the correct synsets for the words in the target language, we find the correct synsets passing in each *Wikipage* and then used that information for mapping words in the target language to the correct synsets.

At first a multilingual, rule based method where monosemous words are used is applied. Then that method is improved with a hypothesis that, if a synset passes in all the translations, then we can say that it is the correct synset. Then 2 graph based multilingual methods are applied.

### 4.1. Bilingual Method - Tagging Page Names, Links and Categories

In this part, page names, links and categories are used to find the synsets of each *Wikipage*. Links are the words that are linked to external pages. To apply this method, we used only target language and English translations of the *Wikipages* together with the bilingual dictionary. At first correct synsets are found for the English using WSD methods and then, they are mapped to the target language.

#### 4.1.1. Tag page name step

In the first step, only page titles are mapped to WordNet synsets. At first, the page titles that cannot have a WordNet synset are removed. For example, there is a page named “Scooby-Doo” in Wikipedia, but it does not exist in the WordNet. Although, it could be useful for extending the current Princeton WordNet[1], as it is not in the scope of this Thesis, such page titles are simply ignored. Only the pages that map to WordNet synsets are extracted. Page



names that have synsets after removing the explanation are also taken into account. Explanations are the words in parenthesis near the page name in some pages (ex: Garfield (the cat)). Then a modified version of the third step in the method of Ponzetto et al. [44] is used. It is explained in the related work. In their method, for WSD purposes, they create disambiguation contexts for the page and for the synsets. To remind, disambiguation context of a page consists of the words in its categories, links and explanations(if exists). Disambiguation context of the synset includes lemmas of its hypernyms, hyponyms, sisterhood and meronyms in the Princeton WordNet[1]. They scored the synsets according to their similarities to the page disambiguation context. Synset with the highest score is selected and mapped to the page name. In this method, we change both page and synset disambiguation contexts by converting them to vector representation using Fasttext Wikipedia embeddings [9].

At first, for the page title a disambiguation context is created. As an example, the page title is “Toy” (Page disambiguation context for page “Toy” is given in figure 4.1.). Possible synsets of that page title in Princeton WordNet [1] are extracted. In this example, they would be: {‘plaything.n.01’, ‘toy.n.02’, ‘toy.n.03’, ‘miniature.n.02’, ‘toy\_dog.n.01’, ‘dally.v.01’, ‘toy.v.02’, ‘play.v.16’}. For each synset, a disambiguation context is created. (For the “Toy” example the synset disambiguation contexts would be like in the table 4.1.)

<p>play, activity, prehistoric, doll, infant, indus, valley, civilization, bow, arrow, mechanical, puzzle, jigsaw, enlightenment, america, dice, board, game, wagon, kite, spinning, wheel, puppet, kaleidoscope, magic, lantern, phantasmagoria, royalty, flora, fauna, production, zoetrope, house, real, wage, angle, girder, gear, nut, bolt, toy, soldier, hollow, casting, tangram, second, world, war, silly, synthetic, rubber, slinky ...</p>
--

**Figure 4.1.** Page disambiguation context for page Toy

After creating disambiguation context for the page and the synsets, those contexts are converted to vector format using Fasttext Wikipedia embeddings [9]. Vector form of the context is calculated by taking the average of the word embeddings of the words inside the context. Given  $n$  and  $s$ , the number of words in the page and synset disambiguation contexts respectively, average embeddings scores of the page and the synset are calculated like in the equations 15 and 16.

**Table 4.1.** Example Synset Disambiguation contexts

<b>“plaything.n.01”</b>	‘americana’, ‘anachronism’, ‘antiquity’, ‘artefact’, ‘article’, ‘artifact’, ‘ball’, ‘balloon’, ‘bear’, ‘block’, ‘board’, ‘building’, ‘button’, ‘catapult’ ...
<b>“toy.n.02”</b>	‘dally’, ‘diddle’, ‘dog’, ‘fiddle’, ‘flirt’, ‘miniature’, ‘play’, ‘plaything’, ‘replica’, ‘replication’, ‘reproduction’, ‘toy’
<b>“toy.n.03”</b>	‘acoustic’, ‘adapter’, ‘adaptor’, ‘aerofoil’, ‘afterburner’, ‘agglomerator’, ‘airfoil’, ‘alarm’, ‘applier’, ‘aspergill’, ‘aspersorium’, ‘asphyxiator’, ‘autocue’ ...
<b>“miniature.n.02”</b>	‘anamorphism’, ‘anamorphosis’, ‘autotype’, ‘carbon’, ‘cast’, ‘casting’, ‘clone’, ‘copy’, ‘dally’, ‘diddle’, ‘facsimile’, ‘fiddle’, ‘flirt’, ‘imitation’, ‘knockoff’ ...
<b>“toy_dog.n.01”</b>	‘barker’, ‘basenji’, ‘belgian’, ‘bow-wow’, ‘brussels’, ‘canis’, ‘corgi’, ‘dally’, ‘dalmatian’, ‘diddle’, ‘dog’, ‘doggie’, ‘doggy’, ‘domestic’, ‘familiaris’, ‘fiddle’, ‘flirt’ ...
<b>“dally.v.01”</b>	‘about’, ‘acquit’, ‘act’, ‘along’, ‘alternate’, ‘antagonise’, ‘antagonize’, ‘anticipate’, ‘approach’, ‘around’, ‘assay’, ‘attack’, ‘attempt’, ‘back’, ‘bear’, ‘begin’, ...
<b>“toy.v.02”</b>	‘control’, ‘dally’, ‘dog’, ‘down’, ‘fiddle’, ‘flirt’, ‘handle’, ‘hands’, ‘knead’, ‘lay’, ‘manage’, ‘manipulate’, ‘massage’, ‘miniature’, ‘monkey’, ‘mouse’, ‘operate’, ‘out’, ‘play’ ...
<b>“play.v.16”</b>	‘act’, ‘backslap’, ‘backwards’, ‘behave’, ‘bend’, ‘bluster’, ‘break’, ‘bungle’, ‘diddle’, ‘dog’, ‘down’, ‘fall’, ‘fiddle’, ‘flirt’, ‘follow’, ‘footle’, ‘freeze’, ‘frivol’ ...

$$v(\text{page}) = \frac{1}{n} \sum_{i=1}^n v_i \quad (15)$$

$$v(\text{synset}) = \frac{1}{s} \sum_{i=1}^s v_i \quad (16)$$

Then, in order to calculate the similarity of the synset with the page name, cosine similarity

between the page vector and the synset vector is calculated (like in equation 17). Synset with the highest score is mapped to the page.

$$score = \frac{v(page) \cdot v(synset)}{\|v(page)\| * \|v(synset)\|} \quad (17)$$

After finding the correct synset for the page name in English page, the next step is to map that synset to the target language. In order to map the found synset to the target language page name, a simple procedure is applied. As the page name in English is the direct translation of the page name in the target language, found synset is directly mapped to the target language page name.

#### **4.1.2. Tag page name words**

Synsets of the page names that has possible synsets in Princeton WordNet [1] are found. Other page names, that does not have synsets in WordNet are simply ignored. In this step, in order to improve the coverage of our methods, words inside those pages are mapped to the correct synsets. At first, page names that are mapped before are ignored. For the rest, all the words inside the page name are extracted. Similarity between the words and their possible synsets are calculated with the same procedure, which is applied in the first method. Synset which has the highest similarity to the page context is selected and mapped to the word.

After finding correct synsets for English, the next is to map them to the target language. In order to map those synsets to target language, bilingual dictionary is used. For each word mapped to a synset, all of its possible translations are taken from the bilingual dictionary and searched in the page title of the target language. If the translation is found, then we can say that it is the translation of the word in the target language. So, the synset is mapped to the target language word.

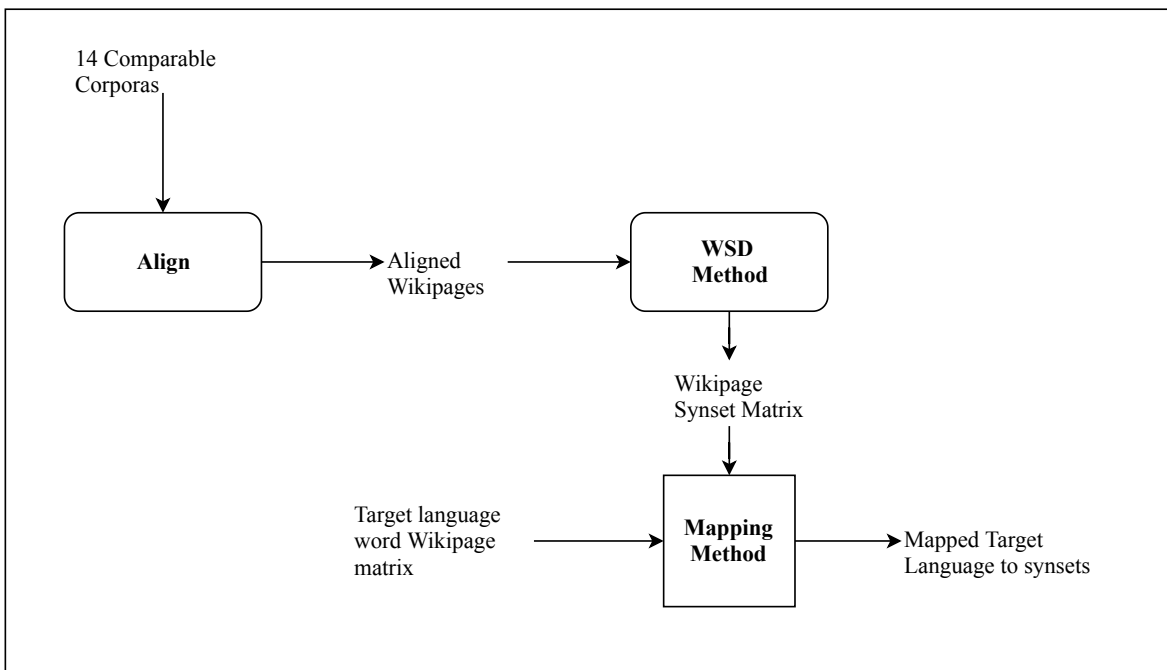
#### **4.1.3. Tag categories and links**

In order to tag the categories and the links, same approach is used. At first, all the words inside the links/categories of the page is extracted. All the links/categories that have translation in the target language links/categories is kept, the rest are ignored. For all the words in the

filtered links/categories, similar approach to the previous steps are applied to find their correct synsets. In order to map those synsets to the target language, translations of the mapped word is searched in the target language links/categories and found word is mapped to the synset.

## 4.2. Multilingual Methods

In this part, applied multilingual methods are explained. Princeton WordNet [1], Wikipedia and bilingual dictionary are used to automatically tag resource poor languages using our multilingual methods. The main idea is to find the correct synsets passing in the *Wikipages* using not only English but also other languages that have a WordNet and then mapping that information to the resource poor language using the interlingual links of Wikipedia. The methods can be divided into two parts. First finding the correct senses occurring in each *Wikipedia* (WSD part) and creating **Wikipedia Synset Matrix(WS)** and second using that matrix together with the term document matrix or in this context **Target Language word Wikipedia Matrix** to map words in the target language to the corresponding Princeton WordNet [1] synsets (Mapping part). The overview of the multilingual methods is given in figure 4.2.



**Figure 4.2.** Multilingual Methods Flow

Two different WSD methods are tried. While first method relies heuristics to disambiguate the synsets, second method uses the graph scoring algorithm to find the intended senses. Both methods use multilinguality of Wikipedia with the basic assumption that if a word is used to refer to a meaning in one language, then a translation of this word in another language is most probably referring to the same meaning. We expect this to be true, at least for senses that are crucial to discuss the main topic in the specific article, as all articles for the same *Wikipage* are written to discuss or describe the same concept.

Before applying the methods, several preprocessing tasks are executed. In order to increase the coverage and to accumulate additional features, articles in 14 languages that have a WordNet are used as the translation source. Comparable corpora between English and those languages are extracted by finding common articles that are associated with an interlingual link.

<sup>1</sup>

An excerpt from Danish-English comparable corpus is given in Figure 4.3.. The article “Boca Juniors” is aligned using the interlingual links available in Wikipedia.

```
<Page>
  <da_page_name>Club Atlético Boca Juniors</da_page_name>
  <en_page_name>Boca Juniors</en_page_name>
  <da_content>
    club, atlético, boca, juniors, argentinsk,
    fodboldklub, der, spiller, landets, bedste,
    primera, division, argentina, klubben, har,
    hjemmebane, bombonera, buenos, aires
  </da_content>
  <en_content>
    club, atlético, boca, juniors, argentine, sports,
    club, based, the, boca, neighborhood, buenos,
    aires, although, many, activities, are, hosted
  </en_content>
</Page>
```

**Figure 4.3.** Comparable corpora between Danish and English

<sup>1</sup>We have used the aligned documents acquired from <https://linguatools.org/tools/corpora/wikipedia-comparable-corpora/>



```

<Wikipage>
  <page_name> Azerbaijan </page_name>
  <French>
    L'Azerbaïdjan, en forme longue la république
    d'Azerbaïdjan, est un pays du Caucase situé
    sur la ligne de division entre l'Europe et
    l'Asie. Sa capitale est Bakou, sa langue
    officielle est l'azéri et sa monnaie est le
    manat.
  </French>
  <English>
    Azerbaijan, officially the Republic of
    Azerbaijan, is a country in the South
    Caucasus region of Eurasia at the
    crossroads of Eastern Europe and
    Western Asia.
  </English>
  <Spanish>
    Azerbaiyán oficialmente República de
    Azerbaiyán para diferenciarla del Azerbaiyán
    iraní es el país soberano más grande en la
    región del Cáucaso, localizado entre Asia
    Occidental y Europa Oriental.
  </Spanish>
</Wikipage>

```

**Figure 4.5.** Example Wikipage with translations to three languages

### 4.3. Rule based Method

In the rule based method a basic approach is taken. This method will be called **Simple Monosemous Method**, for reference in the future. In this method, only synset of monosemous words are selected, polysemous words are ignored. If a word is monosemous, it's synset is added to WS(Wikipage synset matrix). A synset can have multiple meanings in one language, but it could be monosemous in another language. Using multiple source languages increases the chance of disambiguating words using the monosemous sense heuristic. This rule is applied to all *Wikipages* in the corpora.

Using multiple source languages gives rise to a second heuristic. Even if a word is polysemous for all of the languages, if all languages contains a word with a given synset, we assume that this synset is used in this *Wikipedia*. So, we improve the **Simple Monosemous**

**Method**, by adding this assumption. Besides selecting synset of monosemous words, we also select the synsets that take place in all the languages for the given *Wikipedia*. This heuristic method will be called **All Languages Heuristic Method**.

We look at contents of all the translations of the *Wikipedia*. If a word is monosemous, then its synset is added to WS. Possible synsets in language  $l$  of the *Wikipedia* is denoted as  $D_i^l$ . If the word is not a monosemous word, then all of its synsets are added to  $D_i^l$ .

After looking at all the translations, the next step is to find the correct synsets from the possible synsets. We iteratively look at all the polysemous words. Their possible synsets are selected and one that passes in all the languages is selected and added to WS. If more than one synset passes in all the languages, then that word is ignored.

Given the found polysemous synsets of *Wikipedia* as  $S_i$ ,  $L_i$  as the set of languages in the *Wikipedia* and  $D_i^l$  as the possible synsets in language  $l$  of *Wikipedia*, the method is formally given below:

$$S_i = \bigcap_{l \in L_i} D_i^l \quad \forall i \in Wiki \quad (18)$$

To show the tagging process, a toy example will be explained here. For example a page named “Toy” is used (table 4.2.).

At first step, words fra:dont, eng:enjoyable, eng:many, eng:but, eng:airplane etc., are monosemous words, so their synsets are added to WS. For other words, such as “item”, we take all of its synsets from WordNet. They are 'item.n.01', 'detail.n.02', 'item.n.03', 'detail.n.01', 'token.n.01', 'item.r.01'. These synsets are added to the set  $D_{toy}^{english}$ . For example, for French word “jouet”, synsets 'play.v.16', 'toy.v.02', 'toy\_dog.n.01', 'dally.v.01', 'plaything.n.01', 'toy.n.03', 'toy.n.02' are taken from WordNet and added to  $D_{toy}^{french}$ . Then intersection of all languages in  $D_{toy}$  is taken. In this example intersection includes : 'great.s.01', 'game.n.03'. Then looked at all the polysemous words and from their possible synsets one is selected that is in the intersection of all languages in  $D_{toy}$ . As “great.s.01” and “game.n.03” passes in all the languages of the page “Toy” they are selected and added to WS.



**Table 4.2.** Toy example

<b>English</b>	A toy is an item that is used in play, especially one designed for such use. Playing with toys can be an enjoyable means of training young children for life in society. Different materials like wood, clay, paper, and plastic are used to make toys. Many items are designed to serve as toys, but goods produced for other purposes can also be used.
<b>French</b>	Un jouet est un objet dont la fonction principale est de permettre le jeu. Les jouets sont généralement associés avec les enfants ou les animaux domestiques, mais il n'est pas inhabituel pour les adultes et pour certains animaux non-domestiques de jouer avec des jouets.
<b>Spanish</b>	Un juguete es un objeto para jugar, entretener y aprender, generalmente destinado a niños. Ciertos juguetes son apropiados también para animales domésticos, en especialmente perros y gatos, existiendo incluso variedades de juguetes creados específicamente para ellos . Los juguetes pueden ser utilizados individualmente o en combinación con otros

#### 4.4. Graph Methods

Two different graph methods are applied, where both of them used PageRank algorithm [17] but they have different structures, in terms of defining the nodes and edges. First method builds an undirected Graph, but second builds a directed graph. First method builds graphs for each language separately, but second method builds it as a multilingual graph for the *Wikipedia* using all of its translations. That is why, first method will be called Monolingual Graph Method and second Multilingual Graph Method. In addition in the Monolingual Graph method, there is only one edge type, which is between the senses, but in the Multilingual Graph Method there are three different edge types, between words, between words and synsets and between synsets.

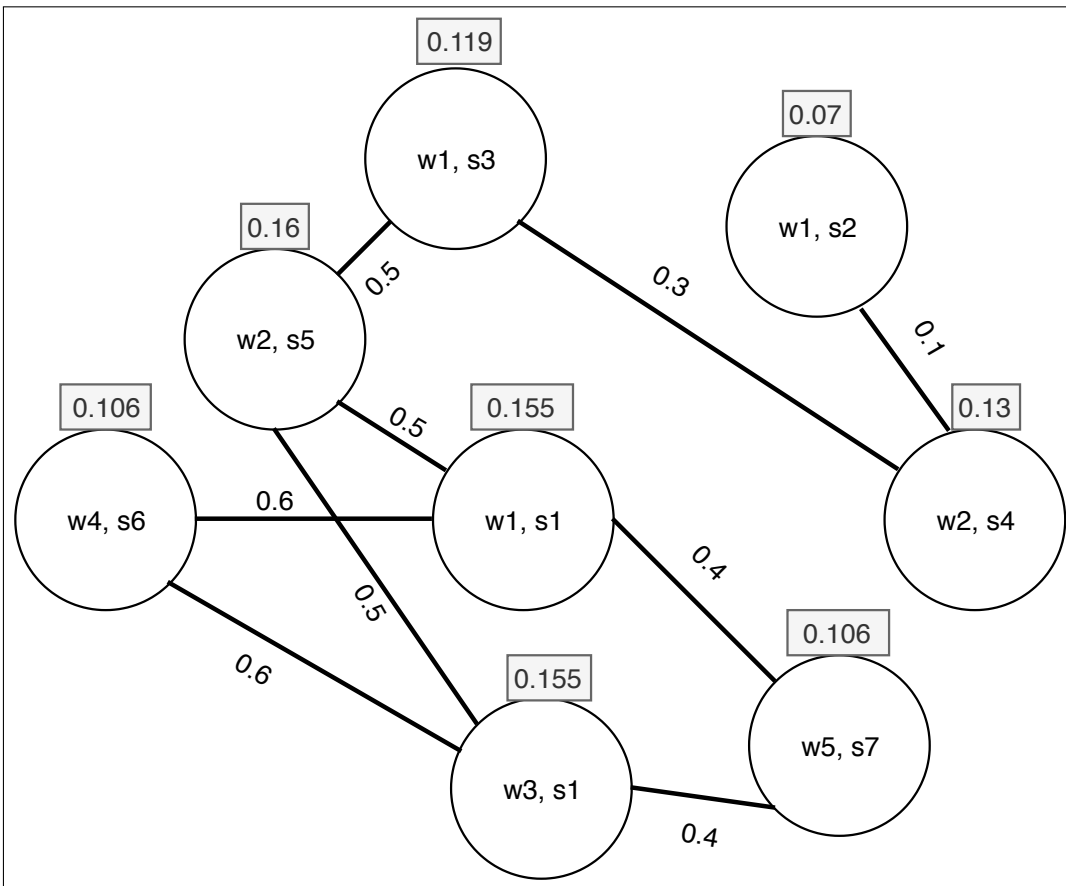
##### 4.4.1. Monolingual Graph Method

We have applied this graph algorithm to each translation inside the *Wikipedia* separately. Vertices of the graph are the senses that take place in that translation of the *Wikipedia*. At first, all the words inside the translation are extracted  $W_l = [w_1, w_2, w_3, w_4, w_5]$ . Each word

has possible synsets.  $S(w_1) = [s_1, s_2, s_3]$  and  $S(w_2) = [s_4, s_5]$ . These word synset pairs are used as the vertices of the graph.

$$V = [(w_1, s_1), (w_1, s_2), (w_1, s_3), (w_2, s_4), (w_2, s_5), (w_3, s_1), (w_4, s_6), (w_5, s_7)]$$

A weighted graph is formed as  $G = (V, E)$  (An example is given in figure 4.6.), where E is the edge set, which is formed using the relations between the synsets of the vertices in the Princeton WordNet [1].



**Figure 4.6.** Monolingual Graph For an example page

Synsets are connected to each other with different relations such as is-A relations hypernymy and hyponymy and part-of relation like meronymy and holonymy etc., in the Princeton WordNet [1]. To add the edges between the nodes, looked at the distance between them in the WordNet. Distance between  $node_i$  and  $node_j$  means the shortest path to start at  $node_i$

and end at  $node_j$ . To both reduce the computational complexity and noise, edges are added between synsets that are connected within a path of length 3. The weights between the nodes are calculated using the inverse of the distances between those nodes. As WordNet is not homogeneous, some synsets have large number of neighbors than others. To decrease its effect to the calculation, weight score is normalized. It is calculated like in equation 19:

$$w_{i,j} = \frac{\frac{1}{d_{i,j}}}{\sum_{s \in WN} \frac{1}{d_{j,s}} * \sum_{s \in WN} \frac{1}{d_{i,s}}} \quad (19)$$

$w_{i,j}$  is the weight between  $node_i$  and  $node_j$ . WN is the set that contains all the synsets in Princeton WordNet [1].  $d_{i,j}$  is the distance between  $node_i$  and  $node_j$  in WordNet.

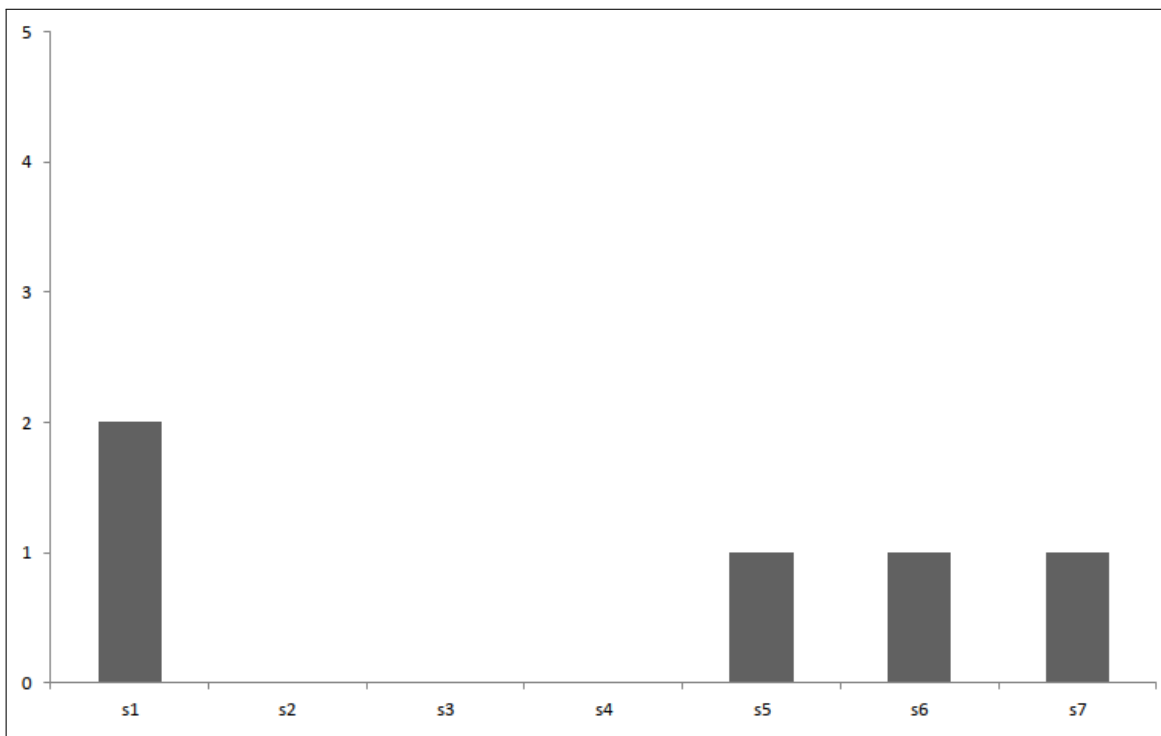
After graph is formed, the next step is to determine how central a synset is in the given article. To find out, different graph scoring methods could be applied. We have applied a local scoring method PageRank [17], where a vertex is scored according to its importance in the graph. As our graph is a weighted graph, weighted PageRank is applied.

$$PR(v) = \frac{1-d}{|V|} + d * \sum_{u \in M(v)} \frac{w(u,v) * PR(u)}{\sum_{j \in M(u)} w(u,j)} \quad (20)$$

The formula can be explained as follows, score of vertex  $v$ ,  $PR(v)$  is calculated.  $d$  is the damping factor that is assigned values between 0 and 1. The original PageRank algorithm [17] uses a damping factor of 0.85. We also use the same damping factor.  $|V|$  is the number of vertices in the graph. Let  $M(v)$  be the set of vertices connected to vertex  $v$  and  $w(u,v)$  is the weight between vertex  $u$  and vertex  $v$ . PageRank scores are iteratively updated using Power iteration method until convergence.

For a word  $w_i$  all vertices that are word-synset pairs  $((w_i, s_*))$  of  $w_i$  are selected as the candidate synsets. The pair with the highest PageRank score is selected. For the example in 4.6., if we look at  $w_1$ ,  $s_1$  is the synset with the highest score, so  $w_1$  is matched with  $s_1$ . For each selected synset, its count is increased by one. So the count of  $s_1$  is increased by 1. After matching all the words to the synsets in figure 4.6., counts of the synsets will be like in the figure 4.7.. This process is applied to all the translations of the *Wikipedia*. For each translation a graph is built and PageRank [17] is applied and correct synsets are found.

Then counts of those synsets are increased by one. After applying this method to all the translations of the *Wikipedia*, the counts of the synsets will be like in figure 4.8..

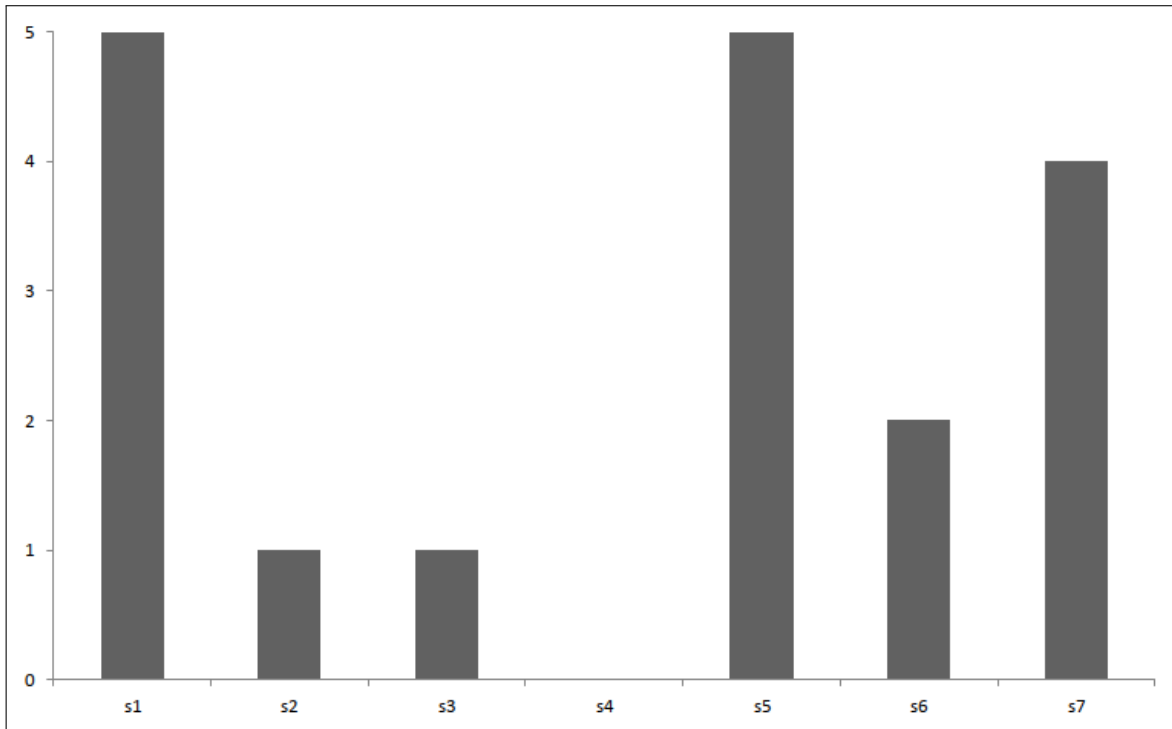


**Figure 4.7.** Synset Counts after applying the graph method to the first translation

After building graph and applying PageRank [17] to all the languages of the *Wikipedia*, each synset will have a count showing how many times it is selected in the *Wikipedia*. In order to find the intended synsets in the *Wikipedia* a final step is applied. This time only English translation of the *Wikipedia* is selected. Again for each word its possible synsets are extracted. Synsets of the words are compared according to the number of times they are selected. Synset with the highest count is selected and added to WS(Wikipedia synset matrix). For the example figure 4.6.,  $w_1$  will be matched with  $s_1$  as from its possible synsets it has the highest count.  $w_2$  will be matched with  $s_5$ ,  $w_3$ ,  $w_4$  and  $w_5$  will be matched with  $s_1$ ,  $s_6$  and  $s_7$  respectively.

#### 4.4.2. Multilingual Graph Method

In this method the information in Wikipedia, WordNet and Wiktionary are integrated to the graph representation. For each *Wikipedia* a separate directed graph is built, where all the translations of that *Wikipedia* are used. In this graph, vertices are the words in the content



**Figure 4.8.** Synset Counts After applying graph algorithm to all the translations

that exist in Wiktionary and all possible synsets of these words in the Princeton WordNet [1]. There are three different edge types, word to word, which shows the relations between the words in the content. Word to sense, which are the edges from a word to all of its possible synsets and sense to sense which is the relation between the synsets in the content.

To add word to word edges, looked at the shortest path distance between the words in a graph which is built using the Wiktionary data. In that graph, vertices are the words in the Wiktionary, for all the possible languages and edges are the translations between the words. If the shortest path distance is higher than threshold (which is selected as 3), no edge is added between two words. Edge weights are calculated as the inverse of the shortest path distance.

Second, we added edges between words and their synsets using Princeton WordNet [1]. For all the possible synsets of the word in Princeton WordNet [1], a directed edge with weight 1 is added.

Finally to calculate the edges between the synset nodes in the graph, Princeton WordNet [1] is used. If the shortest path distance between two synset nodes are below the threshold, then an edge is added between them. The weight of the edge is calculated by taking the inverse of the distance.

After creating the graph, PageRank [17] is applied to the graph, in order to calculate the centrality of the nodes. After applying PageRank [17], the next step is to match the word nodes to the synset nodes. For each English word in the graph, the synset nodes connected to it are selected and the one with the highest PageRank [17] score is matched with the word.

To clarify, how the graph for the multilingual document is created, an example could be useful. To make it simple, only English, Spanish and French will be used. For example a *Wikipage* named “Allobates Wayuu” is selected (which is a frog type).

1. All the languages will be joined and tokenized. Output will be like in figure 4.9. (Some of the words are removed for presentation purposes):

```
eng:tropical, fra:amphibiens, fra:altitude,  
fra:entre, fra:colombie, fra:département,  
eng:dry, eng:eggs, eng:department, eng:natural,  
fra:cope, eng:frog, fra:cette, eng:forest,  
fra:endémique, eng:known, fra:article, eng:amphibian,  
eng:northern, fra:rencontre, fra:amphibia,  
eng:vegetation, eng:tadpole, fra:espèce, eng:family,  
eng:colombia, fra:new, eng:endemic, eng:locality
```

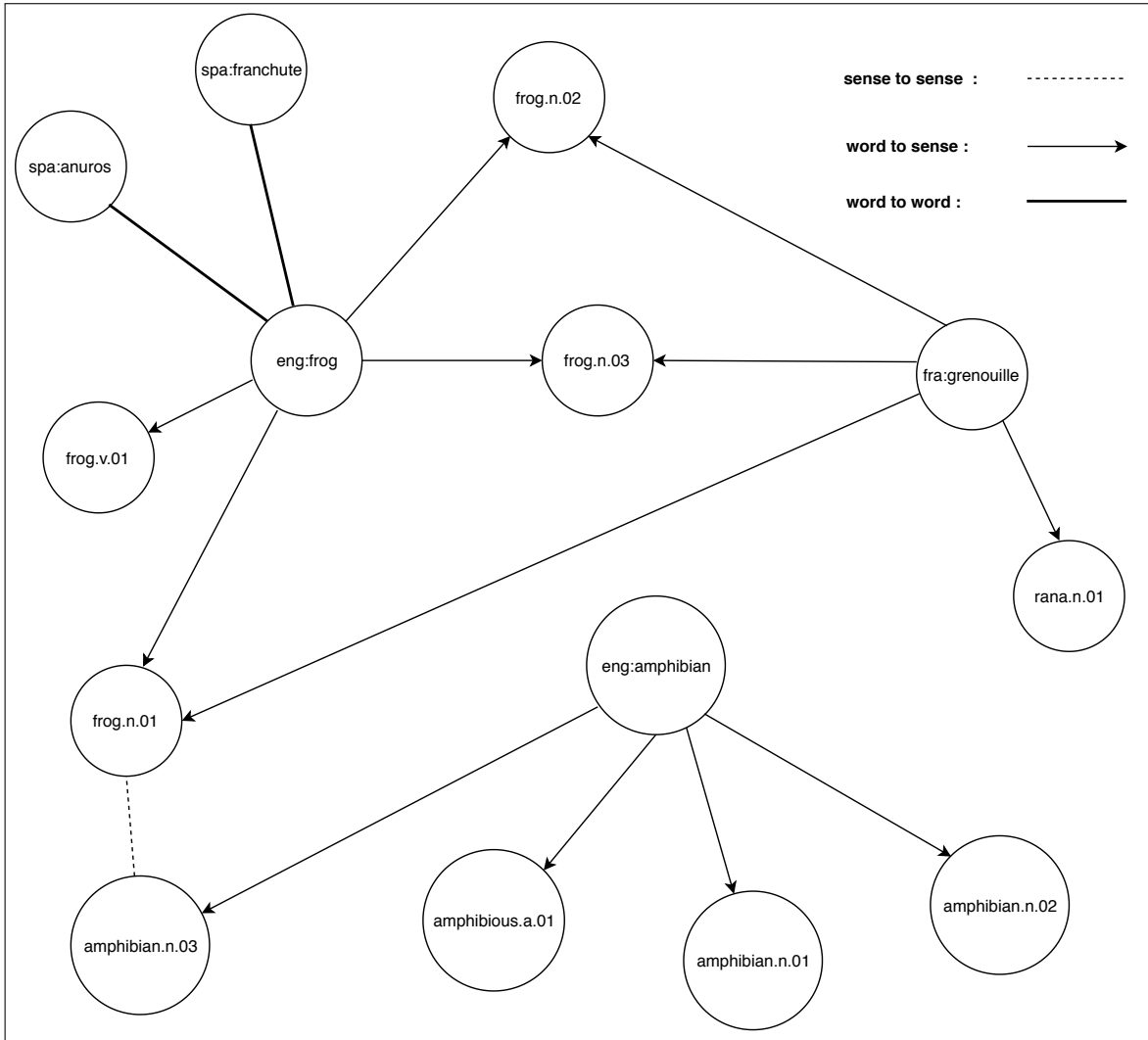
**Figure 4.9.** Multilingual Graph Method input words

2. The words in the format given in figure 4.9. are selected as the vertices to the graph G. In addition their synsets are extracted and they also added as vertices.

3. After adding the edges a graph like in the figure 4.10. is created.

4. Then PageRank algorithm is applied to G. For the English words, their highest scoring synsets are selected and an output is created like in table 4.3.

5. Synsets found are added to the WS (Wikipage Synset Matrix) matrix.



**Figure 4.10.** Multilingual Graph Structure For the toy example

#### 4.5. Mapping target language to correct synsets using Vectorization

All of our multilingual methods generates a Wikipage Synset Matrix as output. In this matrix *Wikipage* ids are rows and synset ids are columns. After finding the correct synsets passing in each *Wikipage*, the next step is to map those synsets to the target language. To do this, a term document matrix should be created for the target language. This matrix is denoted as TD. Then tf-idf is applied to this matrix, in order to decrease effect of common words. Tf-idf stands for term frequency-inverse document frequency, and is calculated as in equation 21

$$w_{t,d} = \log(1 + tf_{t,d}) * \log\left(\frac{N}{df_{t,d}}\right) \quad (21)$$

**Table 4.3.** Multilingual Graph Method output

word	synset
'eng:forest'	'forest.n.01'
'eng:frog'	'frog.n.03'
'eng:locality'	'vicinity.n.01'
'eng:species'	'species.n.01'
'eng:tadpole'	'tadpole.n.01'
'eng:tropical'	'tropical.s.04'
'eng:vegetation'	'vegetation.n.01'

Here  $tf_{t,d}$  stands for the number of times a word  $t$  appears in the *Wikipedia*  $d$ .  $df_{t,d}$  stands for the number of *Wikipages* the word  $t$  takes place and  $N$  is the total number of *Wikipages*.

In order to map words to the synsets, cosine similarity between TD and WS is calculated. The resulting matrix is a mapping between the words in the target language and the synsets. Lets denote it as TS. After building the word synset matrix for the target language, the next step is to filter those synsets to select the correct synsets.

At first, those synsets are sorted according to their score with the word and top  $k$  synsets are selected. This  $k$  value is a tunable parameter (selecting it as 50 gave the best results). After selecting top  $k$  synsets matching for the word, the next step is to use the bilingual dictionary for further filtering. All the English translations of the word are extracted from the bilingual dictionary. Then for each synset, its lemmas are extracted and those lemmas are searched in the English translations. If any lemma of the synset is found in the translation then the synset is matched with the word. If there is none, the word synset match is ignored.

The following example illustrates this algorithm:

Lets select the Turkish as the target language. For demonstration purposes, we assume that there are only 3 *Wikipages* and only 4 words in those *Wikipages*.

1. Wikipage Synset matrix denoted as WS is outputted from the WSD method. A toy output will be used here. It is shown in table 4.4.

**Table 4.4.** Toy Wikipage Synset Matrix

	iron.n.03	iron.n.04	iron.v.01	cast-iron.s.01	iron.n.01	iron.n.02
wikipedia 1	1	0	1	0	0	0
wikipedia 2	0	0	0	0	1	1
wikipedia 3	1	1	0	0	0	0



2. Term document matrix (or word Wikipage matrix in this context) is created for the Turkish translation of Wikipedia. It is denoted as TD. It is shown in table 4.5.

**Table 4.5.** Toy Word Wikipage Matrix

	wikipage 1	wikipage 2	wikipage 3
demir	0	4	1
kapı	3	0	0
anahtar	0	0	2
taş	5	1	0

3. Tf-idf is applied to the Word Wikipage Matrix. The output will be like in the table 4.6.

**Table 4.6.** Tf-idf applied Toy Word Wikipage Matrix

	wikipage 1	wikipage 2	wikipage 3
demir	0	0.12	0.05
kapı	0.29	0	0
anahtar	0	0	0.23
taş	0.14	0.05	0

4. Cosine similarity is applied between the TD and WS and the term synset matrix (or word synset matrix) is created. It is denoted as TS. The resulting matrix is given in table 4.7.

$$TS = \frac{TD \cdot WS}{\|TD\| * \|WS\|} \quad (22)$$

**Table 4.7.** Toy Word Synset Matrix

	iron.n.03	iron.n.04	iron.v.01	cast-iron.s.01	iron.n.01	iron.n.02
demir	0.27	0.38	0	0	0.92	0.92
kapı	0.71	0	1	0	0	0
anahtar	0.71	1	0	0	0	0
taş	0.67	0	0.94	0	0.34	0.34

5. For the rest of the demonstration word “demir” is selected. In order to filter the synsets, at first they are sorted according to their scores. The resulting matrix is given in table 4.8.

**Table 4.8.** Sorted example word synset matrix row

	iron.n.03	iron.n.04	iron.v.01	cast-iron.s.01	iron.n.01	iron.n.02
demir	0.92	0.92	0.38	0.27	0	0

6. k highest scoring synsets are selected. In this example k is selected as 3. So first 3 highest scoring synsets are selected. Output is given in table 4.9.

**Table 4.9.** k highest scoring synsets for word “demir”

	iron.n.03	iron.n.04	iron.v.01
demir	0.92	0.92	0.38

7. After step 6, only 3 synsets are left. In this step, all the translations of the word “demir” to English are extracted using the bilingual dictionary. There is only one translation, which is “iron”. Then lemmas of the synsets are extracted. The lemmas are like [“iron”, “Fe”, “atomic\_number\_26”], [“iron”], [“iron”, “smoothing iron”] respectively for the synsets. So we can see that each of these three synsets contains the word “iron” in their lemmas, so all of them are mapped to the word “demir”.

## 5. EXPERIMENTS AND RESULTS

We perform two different evaluations: first compare our mapping method with a basic mapping method and then we compare our methods with each other and with the baseline methods.

### 5.1. Experimental Setting

In order to make evaluation, the first step is to select the ground truth data. First option to consider is using existing WordNets as ground truth. However, none of the available WordNets are complete. Some words and synsets may be missing in the ground truth. Even if they exist, the link between them might be missing, but when we consider the manual creation procedures of the WordNets, that problem would be less frequent as human annotators tend to associate words they defined with relevant synsets if they both exist. With this assumption, WordNet for German (Germanet) [56] will be used as test dataset. Germanet [56] is chosen as the test dataset, as it has a high WordNet base concept coverage. Germanet [56] contains links to 16,171 PWN [1] synsets involving 18,179 words.

Another option to use for evaluation is to select a set of words and synsets randomly and manually label all possible synset-word pairs as associated or not. This strategy is used by Khodak et al. [27] for Russian and French languages. They randomly selected 200 words per each part of speech (noun, adjective and verb), resulting in a 600 word dataset. Using a machine translation based method candidate synsets are generated for these words. Expert human annotators used these synsets and manually annotated 12,000 candidate word-synset pairs for both French and Russian. Russian dataset is used in our evaluations. When the size of the German and Russian datasets are compared using the number of words, Germanet [56] is 30 times larger and tests a larger portion of the built WordNets.

### 5.2. Evaluation Metrics

To evaluate our methods, standard measures Precision and recall are calculated, by considering the WordNet construction as a word-synset classification task. Micro-average and macro-average are the two different ways to calculate precision and recall scores.

In macro-average method, the results will be calculated for each class separately and then their average will be taken as the overall score. However, in micro-average method, results of all the classes will be aggregated and then average score will be calculated using the total score. In our case, for German dataset micro-average results calculated, but to compare our methods with the results in Khodak et al. [27], macro-average results are calculated for Russian. For German, we evaluated all the word-synset matches and divided the total sum of the correct matchings to the number of word-synset-matches. However, for Russian, we calculated scores for each word separately and then taken their average. As less polysemous words are more frequent, the macro-average values can be higher than micro-average values.

In addition to Precision and Recall we also calculated F1 score. F1 score is a combined measure, which measures the harmonic mean of precision and recall representing both. While precision and recall measures how accurate the assignments are, they do not directly measure how much the built WordNet covers important concepts. For this measure coverage score is calculated to calculate how much our result covers the Core WordNet[57]. Core WordNet [57] first compiled for WordNet 2.0 contains the most important concepts with significant number of edges and centrality. Core WordNet is basically a subset of PWN synsets, formed of 5.000 common and important synsets. A WordNet is expected to cover most of the concepts in Core WordNet and is used to measure how comprehensive a WordNet is.

### **5.3. Baseline Methods**

Three different multilingual methods are evaluated in this study. First one is called Universal WordNet (UWN) [23], that uses SVM method for mapping words to synsets. Another method [24], is called Extended WordNet. In this method basic scoring methods applied for finding the overlap between the definition of the Wiktionary word and gloss of WordNet synset. Third method that we compare with [25], is a multilingual binary classification method, that will be called Synset Expansion Method [25], that uses supervised binary classification method.

## 5.4. Applied Methods

We have applied 1 bilingual and 4 multilingual methods. We will compare the bilingual and multilingual methods separately. In the bilingual method we used bilingual dictionary for mapping from synsets tagged for English to the target language. Our multilingual methods are categorized as rule based and graph based. We have applied 2 rule based and 2 graph based methods. First rule based method is called **Simple Monosemous Method**, and the second one is called **All Languages Method**. First graph method is called **Monolingual Graph method**, although it is a multilingual method, we build graphs for each language separately. However, in our second graph method, we build a graph that contains all the translations of the *Wikipedia*. That is why we called this method **Multilingual Graph Method**.

## 5.5. Results

### 5.5.1. Bilingual Method Evaluation

This section starts with evaluating the result of the Bilingual method using the German dataset as ground truth. In order to see the effectiveness of our mapping method, we first evaluated this method without using our vectorization based mapping method and then used our mapping method. 50 and 10 used as  $k$  values in the mapping method.

**Table 5.1.** Bilingual Method using different Mapping Methods

Method	F1	Precision	Recall	Coverage	Synset Found
Translation Based	39.7	53.06	<b>31.7</b>	<b>67.26</b>	<b>38,553</b>
Vectorization Based top 50	<b>41.37</b>	68.37	29.67	54.33	28,584
Vectorization Based top 10	38.54	<b>75.5</b>	25.9	45.44	25,184

We can see that our mapping method increases the F1 score nearly 2 % and Precision around 15 % with a slight decrease in Recall. It can be said that our mapping method is an improvement compared to a simple Translation based mapping.

### 5.5.2. Multilingual Rule Based Methods Evaluation

We have applied two multilingual rule based methods, Simple Monosemous Method and All Languages method. These methods are evaluated and compared with each other. At first Simple Monosemous Method is evaluated.

**Table 5.2.** Simple Monosemous Method

Top k synsets	F1	Precision	Recall	Coverage	Synset Found
50	<b>44.2</b>	78.3	<b>30.8</b>	<b>54.98</b>	<b>26,382</b>
20	42.1	81.73	28.35	48.61	23,980
10	39.98	<b>84.3</b>	26.2	43.65	22,076

It can be seen that, increasing the  $k$  value, recall scores are increasing, decreasing the precision scores. This is something expected. In our mapping method, we selected  $k$  top synsets and then filtered them using dictionary. When  $k$  is higher, then more synsets are selected as possible synsets. As  $k$  is increased, the scores of new possible synsets are getting smaller and increasing the rate of error.

After looking at the Simple Monosemous Method, we evaluated All Languages Method.

**Table 5.3.** All Languages Method

Top k synsets	F1	Precision	Recall	Coverage	Synset Found
50	<b>44.51</b>	78.16	<b>31.11</b>	<b>56.67</b>	<b>26,684</b>
20	42.35	81.71	28.6	49	24,130
10	40.21	<b>84.7</b>	26.36	44	22,140

Looking at the table, we can see that this method is better than the Simple Monosemous Method in terms of F1 and recall scores and slightly worse for precision. Increase in F1 shows that there were a higher increase in recall but a smaller decrease in precision. (head to head comparison between these methods in given in table 5.4.)

**Table 5.4.** Comparison between our Rule Based Methods  $k = 50$

Method	F1	Precision	Recall	Coverage	Synset Found
Simple Monosemous Method	44.2	<b>78.3</b>	30.8	54.98	26,382
All Languages Method	<b>44.51</b>	78.16	<b>31.11</b>	<b>56.67</b>	<b>26,684</b>

### 5.5.3. Multilingual Graph Based Methods Evaluation

After evaluating the rule based methods, then we have evaluated our graph based methods. We have applied two graph based methods Monolingual Graph method and Multilingual Graph Method. At first, Monolingual Graph method is tested.

**Table 5.5.** Monolingual Graph method

Top k synsets	F1	Precision	Recall	Coverage	Synset Found
50	<b>47.04</b>	72.36	<b>34.85</b>	<b>65.16</b>	<b>30,742</b>
20	44.99	77.3	31.73	57.7	27,349
10	42.87	<b>81.13</b>	29.13	52.06	24,755

Then, Multilingual Graph method is tested. Different two approaches are tried. At first our mapping method is applied to the method where English words are disambiguated. Second, we directly disambiguated the German words in the *Wikipages*. In Multilingual Method, we said that all the languages inside the *Wikipedia* are added to the Graph. So, as we can disambiguate English words in the *Wikipedia*, we can also disambiguate German words. These two methods are compared with each other in table 5.6. We can see that, finding German word synset matches directly has shown worse results.

**Table 5.6.** Multilingual Graph method

Top k synsets	F1	Precision	Recall	Coverage	Synset Found
10	40.49	78.35	27.3	50.6	20,821
50	44.16	69.82	32.3	64.31	25,835
Second approach	40.66	31.86	56.18	96.23	55,157

Then we compared this method with the Monolingual Graph Method in table 5.7. and we saw that, this method gave worse results than the Monolingual Graph method.

**Table 5.7.** Comparison between Graph Methods k = 50

Method	F1	Precision	Recall	Coverage	Synset Found
Monolingual Graph Method	<b>47.04</b>	72.36	<b>34.85</b>	<b>65.16</b>	<b>30,742</b>
Multilingual Graph Method	44.16	69.82	32.3	64.31	25,835

Then in order to see which method has shown best results, we compared all of our methods with each other in table 5.8.. Looking at the table, we can say that Monolingual Graph Method is superior to all our other methods.

**Table 5.8.** Comparison between our Methods k=50

Method	F1	Precision	Recall	Coverage	Synset Found
Bilingual Method	41.37	68.37	29.67	54.33	28,584
Simple Monosemous Method	44.2	<b>78.3</b>	30.8	54.98	26,382
All Languages Method	44.51	78.16	31.11	56.67	26,684
Monolingual Graph Method	<b>47.04</b>	72.36	<b>34.85</b>	<b>65.16</b>	<b>30,742</b>
Multilingual Graph Method	44.16	69.82	32.3	64.31	25,835

#### 5.5.4. Comparison With Baseline Methods

In order to see the performance of our methods, we have compared them with the baseline methods in table 5.9..

**Table 5.9.** Comparison for different methods

Method	F1	Precision	Recall	Coverage	Synset Found
Extended Wordnet	44	76.4	30.9	63.7	19,675
UWN	52.5	59.1	47.3	76.1	50,507
Synset Expansion Method	<b>58.6</b>	76.1	<b>47.6</b>	<b>88.2</b>	<b>54,214</b>
Simple Monosemous Method	44.2	<b>78.3</b>	30.8	54.98	26,382
All Languages Method	44.51	78.16	31.11	56.67	26,684
Monolingual Graph Method	47.04	72.36	34.85	65.16	30,742
Multilingual Graph Method	44.16	69.82	32.3	64.31	25,835

Looking at the results, it can be seen that our methods are better than Extended\_Wordnet method [24] and better than UWN [23] in terms of precision. However, our methods is not nearly as good as the Synset Expansion Method [25]. In that method Wiktionary data is used, which is a lemmatized form, however, we used Wikipedia data for 14 languages. We have applied lemmatization for most of these languages, but only English lemmatizer is a reliable tool, the rest shows not so reliable results.

To further evaluate the methods Russian dataset is used. Also, this time we have used a different evaluation method similar to the method of Khodak et al. [27]. Their calculation of precision is different. They do not punish the method for finding incorrect synset when it is not in the possible synsets of the word. That is why this evaluation method can show different results in terms of precision, compared to the previous method. Also in this method we have evaluated the results for adjectives, nouns and verbs separately, which is used for further depicting the performance of our methods.

Looking at table 5.10. we can see that our Monolingual Graph Method performs better than our rule based method. Also we can see that our methods are good at finding nouns, when compared to other methods. We can see that our graph method is better than UWN [23] and



**Table 5.10.** Comparisons to State-of-the-art in Russian Wordnet Construction

Method	POS	F1-Score	Precision	Recall	Coverage	Synsets
All Languages Method	Adj.	37.56	90.52	27.46	36.7	1,389
	Noun	52.5	<b>94.27</b>	43.1	60.37	12,325
	Verb	31.55	80	21.7	15.67	434
	Total	40.54	88.26	30.75	37.58	14,148
Monolingual Graph Method	Adj.	38.1	<b>91.84</b>	27.63	55.41	2,068
	Noun	54.6	86.64	47.5	75.3	16,147
	Verb	33.94	79.44	23.67	35.2	1,629
	Total	42.21	85.97	32.93	55.3	19,844
Synset Expansion Method (Ercan and Haziyeve, 2019)	Adj.	<b>52.0</b>	87.4	<b>42.4</b>	<b>76.7</b>	7,792
	Noun	<b>60.0</b>	89.2	<b>53.9</b>	<b>85.0</b>	<b>33,491</b>
	Verb	<b>39.6</b>	82.2	<b>29.7</b>	<b>69.0</b>	<b>5,581</b>
	Total	<b>50.5</b>	86.3	<b>42.2</b>	<b>76.9</b>	<b>46,864</b>
UWN (De Melo and Weikum, 2009)	Adj.	38.8	80.3	29.6	51.0	<b>11,412</b>
	Noun	53.0	87.5	45.1	71.1	19,564
	Verb	34.8	74.8	25.7	65.0	3,981
	Total	42.2	80.8	33.4	67.1	30,015
Extended_Wordnet (Bond and Foster, 2013)	Adj.	41.3	91.7	29.2	55.3	2,419
	Noun	53.1	93.5	42.5	68.4	14,968
	Verb	34.8	<b>84.5</b>	23.9	56.6	2,218
	Total	43.1	<b>89.9</b>	31.9	64.2	19,983

Extended\_Wordnet [24] for finding nouns and slightly worse than Synset Expansion method [25]. However, for the adjectives and verbs, we can not say that.

## 5.6. Error analysis

In the experiments, we have compared our methods with each other and with 3 different multilingual baseline methods using 2 different datasets as ground truth. We have seen that our results show comparable results to UWN [23] and Extended\_Wordnet [24], but it is more than 10% worse than the Synset Expansion method [25] in terms of recall. Bilingual dictionary, Wikipedia size of the target language, Wikipedia sizes of the different languages that are used for finding correct synsets in *Wikipages* and quality of stemming and lemmatization are the factors that effects the results of our methods.

We stated that our methods' flow goes like finding correct synsets in the *Wikipages* and then use target language Wikipedia and bilingual dictionary for mapping those synsets to the target language. Finding correct synsets in *Wikipages*, we use the multilinguality of Wikipedia. 14 languages that has WordNets available are used. These languages are given in table 5.11., with the number of Wikipedia articles and WordNet synsets available. It can be seen that some languages has very few number of synsets. When a word is not monosemous in that language but it is missing some synsets and has only one synset available, we will take it as a monosemous word and select the synset as a correct synset. This will increase the false positives in our result. Also it can be seen in table 5.11., that English Wikipedia is very huge compared to others, but using it together with other languages we lose that information

a lot. As we use the multilinguality for detecting signals for the correct synsets, most of the English *Wikipages* will have one or no other translation. So, those pages will be ignored most of the time. This is something that stops our methods to achieve higher recall values.

Another problem is about the size of the Wikipedia of the target language. If the size of the target language Wikipedia is small, recall value will be small. Increasing that size will increase the results. So, this is also a promising thing for the future, as the Wikipedia of target language grows, its recall will also grow.

Another problem we mentioned is the lemmatization and stemming. As Wikipedia is not a lemmatized dataset, we need to apply lemmatization. For most of the languages we were able to apply lemmatization or stemming. However, their qualities are questionable. Even if we match a word with a synset, that word may not be in the correct stem, so it will be ignored.

Finally, we mentioned the extent of the bilingual dictionary that has effect in our results. We applied bilingual dictionary in the mapping process from target language to the *Wikipage* synsets. With increase in the size and quality of the bilingual dictionary, the results will also increase.

**Table 5.11.** Different Wikipedia languages

Language	Number of Articles	Number of synsets in WordNet
English	5,878,258	117,000
Swedish	3,748,139	6,796
French	2,117,762	59,091
Dutch	1,969,922	30,177
Italian	1,536,407	35,001
Spanish	1,528,651	38,512
Polish	1,342,302	33,826
Japanese	1,156,084	57,184
Chinese	1,062,708	42,312
Portuguese	1,008,448	43,895
Finnish	461,066	116,763
Bulgarian	253,061	4,959
Danish	250,398	4,476
Greek	163,800	18,049

## 6. CONCLUSION

### 6.1. Conclusion

In this work, we were able to build WordNets for any language with Wikipedia and bilingual dictionary available. Our method consisted of two parts, where we have found correct synsets for the Wikipages in the first part and in the second part mapped those synsets to the target language. We do not need to run the first part of our method each time. It is only needed to run once. Then for any language that we want to build a WordNet the mapping step is applied. It is stated in this work that using a more complicated WSD method will improve the results. We started with a method, where we have applied a rule based monosemous method. Then we added more rules to that method and achieved better results. Then we applied a graph algorithm and found out that this method achieves a further improvement. So, it is proven that a better WSD approach improves the results. It is believed that, in the future using a more complex WSD method would improve the results further.

### 6.2. Future Research Directions

In this project, we were able to build WordNet using Wikipedia and bilingual dictionary and found fair results. However, as we stated before, we believe that improving the WSD method we can improve our results. One approach to try is to use a supervised method that is also good at all words task. We saw that our graph method was an improvement, but it needs further improvement. For example, weighting applied to the edges can be changed, because it was straightforward. We were taking the inverse of the distance in the WordNet and then normalized it with the sum of the scores of source edge and target edge.

## REFERENCES

- [1] George A Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J Miller. Introduction to wordnet: An on-line lexical database. *International journal of lexicography*, 3(4):235–244, **1990**.
- [2] Piek Vossen. Introduction to eurowordnet. In *EuroWordNet: A multilingual database with lexical semantic networks*, pages 1–17. Springer, **1998**.
- [3] Dan Tufis, Dan Cristea, and Sofia Stamou. Balkanet: Aims, methods, results and perspectives. a general overview. *Romanian Journal of Information science and technology*, 7(1-2):9–43, **2004**.
- [4] Piek Vossen. A multilingual database with lexical semantic networks. *Dordrecht: Kluwer Academic Publishers*. doi, 10:978–94, **1998**.
- [5] Zellig S Harris. Distributional structure. *Word*, 10(2-3):146–162, **1954**.
- [6] Thomas K Landauer, Peter W Foltz, and Darrell Laham. An introduction to latent semantic analysis. *Discourse processes*, 25(2-3):259–284, **1998**.
- [7] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, **2013**.
- [8] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543. **2014**.
- [9] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, **2017**.
- [10] Henry Kučera and Winthrop Nelson Francis. *Computational analysis of present-day American English*. Dartmouth Publishing Group, **1967**.
- [11] Douglas B Paul and Janet M Baker. The design for the wall street journal-based csr corpus. In *Proceedings of the workshop on Speech and Natural Language*, pages 357–362. Association for Computational Linguistics, **1992**.

- [12] Philip Resnik. Selectional preference and sense disambiguation. *Tagging Text with Lexical Semantics: Why, What, and How?*, **1997**.
- [13] Linton C Freeman. Centrality in social networks conceptual clarification. *Social networks*, 1(3):215–239, **1978**.
- [14] Phillip Bonacich and Paulette Lloyd. Eigenvector-like measures of centrality for asymmetric relations. *Social networks*, 23(3):191–201, **2001**.
- [15] Stephen P Borgatti. *The key player problem*. na, **2003**.
- [16] Stanley Wasserman and Katherine Faust. *Social network analysis: Methods and applications*, volume 8. Cambridge university press, **1994**.
- [17] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, **1999**.
- [18] David Gibson, Jon Kleinberg, and Prabhakar Raghavan. Inferring web communities from link topology. In *Proceedings of the ninth ACM conference on Hypertext and hypermedia: links, objects, time and space—structure in hypermedia systems: links, objects, time and space—structure in hypermedia systems*, pages 225–234. Citeseer, **1998**.
- [19] Khang Nhut Lam, Feras Al Tarouti, and Jugal Kalita. Automatically constructing wordnet synsets. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 106–111. **2014**.
- [20] Benoît Sagot and Darja Fišer. Building a free french wordnet from multilingual resources. In *OntoLex*. **2008**.
- [21] Nasrin Taghizadeh and Hesham Faili. Automatic wordnet development for low-resource languages using cross-lingual wsd. *Journal of Artificial Intelligence Research*, 56:61–87, **2016**.
- [22] Patanakul Sathapornrungskij and Charnyote Pluempitiwiriyawej. Construction of thai wordnet lexical database from machine readable dictionaries. *Proc. 10th Machine Translation Summit, Phuket, Thailand*, **2005**.

- [23] Gerard De Melo and Gerhard Weikum. Towards a universal wordnet by learning from combined evidence. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 513–522. ACM, **2009**.
- [24] Foster R. Bond, F. Linking and extending an open multilingual wordnet. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 1352–1362. **2013**.
- [25] Gonenc Ercan and Farid Haziyeu. Synset expansion on translation graph for automatic wordnet construction. *Information Processing & Management*, 56(1):130–150, **2019**.
- [26] Jugal Kalita et al. Enhancing automatic wordnet construction using word embeddings. In *Proceedings of the Workshop on Multilingual and Cross-lingual Methods in NLP*, pages 30–34. **2016**.
- [27] Mikhail Khodak, Andrej Risteski, Christiane Fellbaum, and Sanjeev Arora. Automated wordnet construction using word embeddings. In *Proceedings of the 1st Workshop on Sense, Concept and Entity Representations and their Applications*, pages 12–23. **2017**.
- [28] Sanjeev Arora, Yingyu Liang, and Tengyu Ma. A simple but tough-to-beat baseline for sentence embeddings. **2016**.
- [29] Heidi Sand, Erik Velldal, and Lilja Øvrelid. Wordnet extension via word embeddings: Experiments on the norwegian wordnet. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 298–302. **2017**.
- [30] Antoni Oliver and Salvador Climent. Automatic creation of wordnets from parallel corpora. In *LREC*, pages 1112–1116. **2014**.
- [31] Lluís Padró, Samuel Reese, Eneko Agirre, and Aitor Soroa. Semantic services in freeling 2.1: Wordnet and ukb. In *5th Global WordNet Conference*, pages 99–105. **2010**.
- [32] Martin Saveski and Igor Trajkovski. Automatic construction of wordnets by using machine translation and language modeling. In *13th Multiconference Information Society, Ljubljana, Slovenia*. **2010**.

- [33] Rudi L Cilibrasi and Paul MB Vitanyi. The google similarity distance. *IEEE Transactions on knowledge and data engineering*, 19(3):370–383, **2007**.
- [34] Daniel M Bikel. Automatic wordnet mapping using word sense disambiguation. In *2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*. **2000**.
- [35] Zahra Mousavi and Heshaam Faili. Persian wordnet construction using supervised learning. *arXiv preprint arXiv:1704.03223*, **2017**.
- [36] Michael Lesk. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proceedings of the 5th annual international conference on Systems documentation*, pages 24–26. Cite-seer, **1986**.
- [37] Jim Cowie, Joe Guthrie, and Louise Guthrie. Lexical disambiguation using simulated annealing. In *Proceedings of the 14th conference on Computational linguistics-Volume 1*, pages 359–365. Association for Computational Linguistics, **1992**.
- [38] Scott Kirkpatrick, C Daniel Gelatt, and Mario P Vecchi. Optimization by simulated annealing. *science*, 220(4598):671–680, **1983**.
- [39] Adam Kilgarriff and Joseph Rosenzweig. Framework and results for english senseval. *Computers and the Humanities*, 34(1-2):15–48, **2000**.
- [40] Satanjeev Banerjee and Ted Pedersen. An adapted lesk algorithm for word sense disambiguation using wordnet. In *International conference on intelligent text processing and computational linguistics*, pages 136–145. Springer, **2002**.
- [41] Rada Mihalcea. Using wikipedia for automatic word sense disambiguation. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 196–203. **2007**.
- [42] Roberto Navigli and Mirella Lapata. An experimental study of graph connectivity for unsupervised word sense disambiguation. *IEEE transactions on pattern analysis and machine intelligence*, 32(4):678–692, **2009**.

- [43] Roberto Navigli. Semi-automatic extension of large-scale linguistic knowledge bases. In *FLAIRS Conference*, pages 548–553. **2005**.
- [44] Simone Paolo Ponzetto and Roberto Navigli. Knowledge-rich word sense disambiguation rivaling supervised systems. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 1522–1531. Association for Computational Linguistics, **2010**.
- [45] Rada Mihalcea. Unsupervised large-vocabulary word sense disambiguation with graph-based algorithms for sequence data labeling. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*. **2005**.
- [46] Eneko Agirre and Aitor Soroa. Personalizing pagerank for word sense disambiguation. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 33–41. Association for Computational Linguistics, **2009**.
- [47] Mikael Kågeback and Hans Salomonsson. Word sense disambiguation using a bidirectional lstm. *arXiv preprint arXiv:1606.03568*, **2016**.
- [48] Tommaso Pasini and Roberto Navigli. Train-o-matic: Large-scale supervised word sense disambiguation in multiple languages without manual training data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 78–88. **2017**.
- [49] Roberto Navigli and Simone Paolo Ponzetto. Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250, **2012**.
- [50] Dayu Yuan, Julian Richardson, Ryan Doherty, Colin Evans, and Eric Altendorf. Semi-supervised word sense disambiguation with neural models. *arXiv preprint arXiv:1603.07012*, **2016**.
- [51] Bharath Dandala, Rada Mihalcea, and Razvan Bunescu. Multilingual word sense disambiguation using wikipedia. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 498–506. **2013**.



- [52] Bharath Dandala, Rada Mihalcea, and Razvan Bunescu. Word sense disambiguation using wikipedia. In *The People's Web Meets NLP*, pages 241–262. Springer, **2013**.
- [53] Roberto Navigli, David Jurgens, and Daniele Vannella. Semeval-2013 task 12: Multilingual word sense disambiguation. In *Second Joint Conference on Lexical and Computational Semantics (\*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, volume 2, pages 222–231. **2013**.
- [54] Roberto Navigli and Simone Paolo Ponzetto. Babelnet: Building a very large multilingual semantic network. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 216–225. Association for Computational Linguistics, **2010**.
- [55] Martin F Porter. Snowball: A language for stemming algorithms, **2001**.
- [56] Piek Vossen, Geert Adriaens, Nicoletta Calzolari, Antonio Sanfilippo, and Yorick Wilks. Automatic information extraction and building of lexical semantic resources for nlp applications. In *Proceedings of the ACL/EACL-97 workshop, Madrid*. **1997**.
- [57] Jordan Boyd-Graber, Christiane Fellbaum, Daniel Osherson, and Robert Schapire. Adding dense, weighted connections to wordnet. In *Proceedings of the third international WordNet conference*, pages 29–36. Citeseer, **2006**.



HACETTEPE UNIVERSITY  
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING  
THESIS/DISSERTATION ORIGINALITY REPORT

HACETTEPE UNIVERSITY  
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING  
TO THE DEPARTMENT OF Computer Engineering

Date: 27.06.2019

Thesis Title / Topic: Automatic WordNet Construction Using Wikipedia Data

According to the originality report obtained by myself/my thesis advisor by using the Turnitin plagiarism detection software and by applying the filtering options stated below on 27/06/2019 for the total of 62 pages including the a) Title Page, b) Introduction, c) Main Chapters, d) Conclusion sections of my thesis entitled as above, the similarity index of my thesis is 9 %.

Filtering options applied:

1. Bibliography/Works Cited excluded
2. Quotes excluded
3. Match size up to 5 words excluded

I declare that I have carefully read Hacettepe University Graduate School of Science and Engineering Guidelines for Obtaining and Using Thesis Originality Reports; that according to the maximum similarity index values specified in the Guidelines, my thesis does not include any form of plagiarism; that in any future detection of possible infringement of the regulations I accept all legal responsibility; and that all the information I have provided is correct to the best of my knowledge.

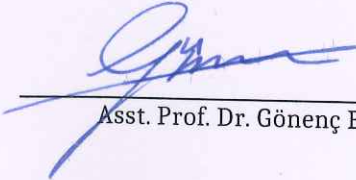
I respectfully submit this for approval.

27.06.2019  
Haziye  
Date and Signature

Name Surname: Farid Haziyevev  
Student No: N16121459  
Department: Computer Engineering  
Program: Computer Engineering  
Status:  Masters  Ph.D.  Integrated Ph.D.

**ADVISOR APPROVAL**

APPROVED.

  
Asst. Prof. Dr. Gonenç Ercan

# CURRICULUM VITAE

## Credentials

Name,Surname: Farid HAZIYEV  
Place of Birth: Sumgait,Azerbaijan  
Marital Status: Single  
E-mail: ferid.heziyev@gmail.com  
Address: Computer Engineering Dept., Hacettepe University  
Beytepe-ANKARA

## Education

BSc. : Industrial Engineering Dept.,  
Middle East Technical University, Turkey

## Foreign Languages

English  
Russian

## Work Experience

Software Developer (2015-2017)  
Machine Learning Engineer (2018-Present)

## Areas of Experiences

Machine Learning, NLP

## Project and Budgets

-

## Oral and Poster Presentations

-