eman ta zabal zazu

Universidad
del País Vasco

Euskal Herriko
Unibertsitatea

Coping with Data Scarcity: First Steps towards Word Expansion for a Chatbot in the Urban transportation Domain

Author: Eneritz García-Montero

Advisors: Arantza del Pozo-Echezarreta and Itziar González-Dios

hap/lap

Hizkuntzaren Azterketa eta Prozesamendua
Language Analysis and Processing

Final Thesis

September 2020

Departments: Computer Systems and Languages, Computational Architectures and Technologies, Computational Science and Artificial Intelligence, Basque Language and Communication, Communications Engineer.

Laburpena

Hizkuntzaren Prozesamenduan (HP) zenbait arlotan hitzak erabili izan dira tradizionalki zabaltze-tekniken garapenean, hala nola Informazioaren Berreskurapenean (IB) edota Galdera-Erantzun (GE) sistemetan. Master tesi honek bi hurbilpen aurkezten ditu Elkarrizketa-Sistemen (ES) arloan zabaltze-teknikak garatze aldera, zehazkiago Donostiako (Gipuzkoa) hiri-garraiorako chatbot baten ulertze-modulua garatzera zuzendurik. Lehenengo hurbilpenak hitz-bektoreak erabiltzen ditu semantikoki antzekoak diren terminoak erauzteko, kasu honetan FastText-eko aurre-entreinaturiko embedding sorta espainieraz eta bigarren hurbiltzeak hitzen adiera-desanbiguazioa erabiltzen du sinonimoak datu-base lexiko baten bidez erauzteko, kasu honetan espainierazko WordNet-a. Horretarako, ataza kolaboratibo bat diseinatu da, non corpusa osatuko baitugu balizko-egoera erreal baten sarrerak jasoz. Bestalde, domeinuz kanpo dauden sarrerak identifikatze aldera, bi esperimentu sorta garatu dira. Lehenengo fasean kalifikatze sistema bat garatu da, non corpuseko terminoak Term Frequency-Inverse Document Frequency (TF-IDF) erabiliz ordenatzen baitiren eta ondoren kalifikatze-sistema kosinu-antzekotasunaren bidez osatzen da. Bigarren faseak aurreko kalifikatze-sistema formalizatuko da, hiru datu-multzo prestatuz eta estratifikatuz. Datu-multzo hauek erregresore lineal bat eta Kernel linealarekin euskarri bektoredun makina bat entreinatzeko erabili dira. Emaitzen arabera, aurre-entreinaturiko bektoreek leialtasun handiagoa daukate input errealari dagokionez. Hala ere, datu-base lexikoek estaldura linguistiko zabalagoa gehituko diote zabalduriko corpus hipotetikoari. Azkenik, domeinuaren diskriminazioari dagokionez, emaitzek TF-IDF-tik erauzitako termino gehienen zeukan datu-multzoa hobesten dute.

Abstract

Text expansion techniques have been used in some subfields of Natural Language Processing (NLP) such as Information Retrieval or Question-Answering Systems. This Master's Thesis presents two approaches for expansion within the context of Dialogue Systems (DS), more precisely for the Natural Language Understanding (NLU) module of a chatbot for the urban transportation domain in San Sebastian (Gipuzkoa). The first approach uses word vectors to obtain semantically similar terms while the second one involves synonym extraction from a lexical database. For this purpose, a corpus composed of real case scenario inputs has been exploited. Furthermore, the qualitative analysis of the implemented expansion techniques revealed a need to filter out-of-domain inputs. In relation to this problem, two different sets of experiments have been carried out. First, the feasibility of using Term Frequency-Inverse Document Frequency (TF-IDF) and cosine similarity as discrimination features was explored. Then, linear regression and Support Vector Machine (SVM) classifiers were trained and tested. Results show that pre-trained word embedding expansion constitutes a more loyal representation of real case scenario inputs, whereas lexical database expansion adds a wider linguistic coverage to a hypothetically expanded version of the corpus. For out-of-domain detection, increasing the number of features improves both, linear regression and SVM classification results.

Acknowledgements

First and foremost, my gratitude goes to my fellow colleagues in Vicomtech, specially for the Dialogue Systems line, where Laura García-Sardiña and Manex Serras-Sáenz have been a fundamental role in my development as a student and a worker. It was with you that I carried out my first experiments when I finished my Master's Degree, and thanks to your endless patience and your Socratic method I have been able to go forward, regardless of the difficulties. Needless to say, I would also like to thank Arantza del Pozo-Echezarreta, my advisor in Vicomtech, whose empathy and experience have pushed me along the way to become a better student and worker.

Next, I would like to thank my advisor from the Master's Degree, Itziar González-Dios, for such a dedicated work with my thesis. It is not always easy to understand a student's confused explanations, nor it is to give the right answer each time. Thank you for cheering me up when I felt like I was not doing my best.

Last but not least, I owe more than a few lines for all my friend who have stood by my side every day. First, to my partner in crime, Salvador Lima-López, thank you for being so patient and caring, even when I did not want to listen. You're a role model to me. Second, to my roommates, Mikel Quintana-Uriarte, Jon Mikel Olmos-Serna and Ander González-Docasal, for making me laugh and cry, specially when I did not deserve your time. You made me the luckiest person in the world while I lived with you. Third, to my partner in life, Pepe Burgos-Ruiz, whose endless sense of humour and sarcasm never let me down. Thank you for listening to all my (involuntary) monologues. Finally, to my mum, whose endless wisdom and sense of humour are a cornerstone for me.

Contents

List of Figures

List of Tables

1 Introduction

Over the past few years, Dialogue Systems (DS) have been gaining increasing attention from both the scientific and the industrial communities. Along with the recent introduction of artificial neural networks and Deep Learning techniques, human-computer interactions have become a plausible reality and a highly requested utility within the Natural Language Processing (NLP) field. From a historical point of view, the conversational systems have evolved over the course of the years due to the development of computer systems, more sophisticated theories of dialogue phenomena, and the increased need in commercial applications (Jokinen, 2000).

Traditional DS assist the user to complete a certain task. In those cases, the dialogue response is typically represented as a module pipeline that includes four key components: Natural Language Understanding (NLU), Dialogue State Tracker, Dialogue Policy Learning and Natural Language Generation (NLG). In (1) the four of them are depicted in a visual diagram.

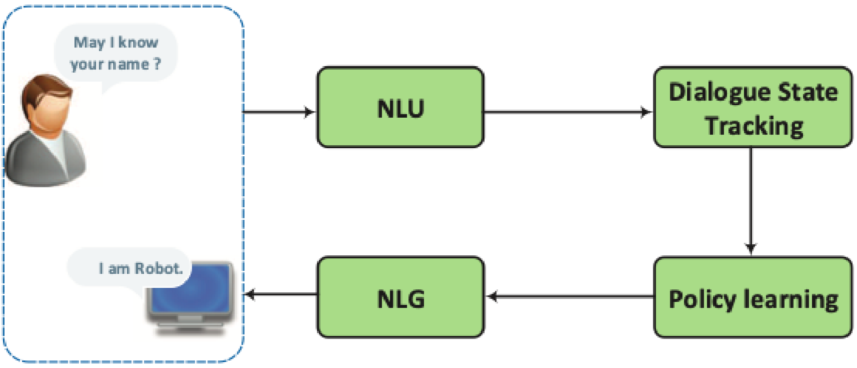

Figure 1: Modules involved in a Task-Oriented Dialogue System (Chen et al., 2017)

The first component of a task-oriented DS is the NLU. NLU is a task that typically extracts structured semantic knowledge from a text. This task facilitates analyzing and understanding relationships between unstructured texts and their corresponding semantic interpretation (Jung, 2019). The concept of semantic interpretation refers to a set of slot-value representations, manually created in order to represent natural language in computational terms. Slot-value representations are predefined according to different communication scenarios with the aim of covering every possible user input. The fact that these slots are handcrafted makes the semantic interpretation step more time-consuming than others involved in the process of the creation of a NLU.

There are mainly two steps involved in NLU. The first of them is intent detection, where the module processes inputs and the utterance is classified into one of the predefined intents. The second one is word information extraction or slot-filling. During this process, the words in the sentence are assigned with semantic labels (Chen et al., 2017). Unlike intent detection, slot filling is usually defined as a sequence labeling problem. That is, given a sentence input, the output will be transformed into a value-slot sequence. Another way of picturing an NLU module would be a language comprehension module trained with a set of hand-crafted examples that are fed to the module. In this manner, it becomes able to classify in a robust way.

This Master's Thesis was developed inside the framework of a project. This project aimed to develop a chatbot for the urban transportation domain, which consisted on addressing user inquiries regarding the *MUGI* transportation card used all over the province of Gipuzkoa. That being so, two drawbacks were identified about developing a task-oriented Dialogue System (DS): data scarcity and time-consuming manual work, involved in gathering and labeling data to train the NLU. In order to solve the problems they pose, this Master's Thesis explores methodologies in which word expansion can help alleviate these two issues.

After the NLU module, the Dialogue State Tracker is used. By means of this component, the user's intention is tracked in every turn of the conversation, maintaining a distribution over multiple hypotheses of the true dialogue state (Chen et al., 2017), typically relying on hand-crafted rules in order to obtain the most likely result.

The third component is the Policy Learning State module. This module uses the representation output from the state tracker, and it is in charge of generating the next system action. In other words, Policy Learning is a choice of actions for a given internal system state (Thomson and Young, 2010). Due to this reason, this state is dependant on the Dialogue State Tracking module.

The last component is the NLG module. It generates natural language utterances out of system actions given by the policy learner. The outcome results of this phase are usually referred to as *variation*, and it is usually evaluated in terms of adequacy, fluency and readability (Stent et al., 2005), as it provides the user with a response that should be as close to a real conversation as possible. Researchers aim to provide the user with a natural interaction, avoiding issues such as ambiguity or mistaken outputs.

Conventional NLG modules divide the task into sentence planning and surface realisation: the former maps semantic symbols into an intermediary representation of the utterance and the latter converts that intermediate form into the final text (Wen et al., 2015). Nowadays, the most widely used approach are rule-based statistical approaches, such as generating the most-likely context-free derivations given a corpus (Belz, 2008), for instance. These pre-designed, handcrafted rules are specifically designed to cover a certain range of entities that belong to a specific domain, often becoming time-consuming and difficult to adapt to different domains.

This thesis is organized as follows: Chapter 2 explains the related work for expansion in NLP, along with a description of the resources used. The Chapter 3 is about the processes the corpus underwent and Chapter 4 and 5 explain the experimentation phase in which the corpus was used, namely for expansion and out-of-domain detection purposes. Finally, Chapter 6 concludes this work and provides some insight into possible future developments.

2 Related Work

This section introduces the resources used in this Master's Thesis, which involve two different types of materials: first the use of word embeddings for expansion and second, the use of words for synonym extraction from lexical databases.

2.1 Resources

The first experimental phase of this Master's Thesis involves the use of word embeddings for expansion. In this work, pretrained embeddings are utilized. Precisely, FastText (Grave et al., 2018) pretrained embeddings. These embeddings are widely used in the research community, as they provide researchers with a considerable amount of data in a simple way, as well as being language-specific, with pre-trained models for up to 157 languages. Furthermore, FastText's embeddings have been trained on Wikipedia[1] and Commoncrawl (Crawl, 2019) and they contain a vast text-collection.

The second experimental phase involves the use of databases for synonym extraction. There are two types of sources that can be used: Knowledge-Based (KB) sources and Corpus-Based (CB) approaches. A KB is a hierarchical and complex database containing structured and unstructured data. An example of this type of representations would be an ontology, as it contains classes, subclasses and instances. Thus, in the case of NLP, KB systems involve necessarily the use of resources such as WordNet (Miller, 1998) or BabelNet (Navigli and Ponzetto, 2010). In this case it will only be WordNet. WordNet is a lexical database initially designed for the English language, and later adapted to other languages such as Spanish or French. It can be thought of as a large electronic dictionary, as it contains information about some 155,000 nouns, verbs, adjectives, and adverbs (Miller, 1998). Here, semantically similar terms are interlinked, and this semantic network constitutes the arrangement of the whole database. In the context of this work, this resource is used in order to retrieve not just synonyms or semantically similar terms, but also hyperonyms and hyponyms, enriching the resulting expanded corpus with a wider variety of options.

Both resources have different strenghts. Regarding FastText, they are expected to have many different representations for maybe even the same entity, but written in a different way each time. As mentioned in this chapter, they have been trained on Commoncrawl and Wikipedia, therefore it is possible that there exist all types of versions for the same word coming from Commoncrawl. After all, it comes from a huge source that contains everything possible every written online and that includes, orthographic errors or special characters, for instance. As for WordNet, on the contrary, it is more probable for it to be clean and precise. It contains lexical elements that belong strictly to the language, which makes it a good linguistic support in terms of synonymy.

[1] https://en.wikipedia.org/wiki/Main_Page

2.2 Word Embeddings for Expansion

There are various fields in NLP that have utilized word embedding's properties for expansion purposes such as Information Retrieval or text classification. Word-vectors —also known as distributed word representations—capture precise syntactic and semantic word-relationships. By means of mathematical operations, word embeddings are trained to represent terms with dense, low-dimensional and real-valued vectors (Wang et al., 2016). In the geometric embedding space similar words tend to cluster with each other, so that semantically resembling terms obtain nearby vectorial representations. These representations contain linguistic regularities and patterns. They can be represented as linear translations, where $vec(''Madrid'') - vec(''Spain'') + vec(''France'')$ should be closer to $vec(''Paris'')$ than any other vectors. Similarly, the analogy a:b c:d shows the same concept, where d is unknown, embedding vectors Xa, Xb, Xc are found and $y = Xb - Xa + Xb$ is computed, where y is the continuous space representation of the best answer (Mikolov et al., 2013).

State-of-the-art techniques for expansion focus on term set expansion, where they return the k nearest neighbors around the seed terms as the expanded set. These terms are represented by their co-occurrence companion or embedding vector in a training corpus (Mamou et al., 2018). For instance, in Mamou et al. (2018) explain that given the term *Python*, it is preferable for the word to be expanded to other programming languages, rather than other concepts related to the field such as *code* or *debugging*.

Additionally, there are other research lines in NLP that benefit from the advantages of expansion. One of those fields is answer passage retrieval. Document or answer passage retrieval is used as the first step in Question Answering (QA) systems, where some studies carry out query expansion techniques. For example, Roy (2019) worked on the automatic or semi-automatic addition of terms to the original query to close the semantic gap between a user's query and the information need from the system.

In the case of Wang et al. (2016), they find it difficult to classify short texts due to the lack of context that they provide to the systems. Traditionally, text classification methodologies expand short texts using Latent Semantics, first training language models and secondly performing semantic composition to obtain phrase level representations. Knowing that in embedding spaces similar words tend to cluster together, Wang et al. (2016) tried to overcome data scarcity problems for short text classification by combining word embedding clustering and Convolutional Neural networks. Similarly, Phan et al. (2008) presented a general framework to deal with short and sparse text expansion, utilizing hidden topic names retrieved from Wikipedia via Latent Dirichlet Allocation (LDA). This is also the case for Dong et al. (2020), where they explore four different strategies for extending sparse text that are influenced by domain knowledge for short text classification. In this work they use enriched textual data to predict donations by means of machine learning techniques.

Regarding Information Retrieval (IR), there are some studies that show the need to apply various expansion techniques. This is the case for Song et al. (2007), where they propose a novel semantic query expansion technique that combines the use of a structured lexical database, namely WordNet (Miller, 1998), along with NLP techniques and association rules. Additionally, Colace et al. (2015) propose a query expansion method that automatically extracts a set of Weighted Word Pairs from a set of topic-related documents. Their technique relies on explicit relevance feedback, which consists of receiving feedback documents selected by the user himself that contain topic-related terms.

3 Corpus Description

The context of this work is linked to the development of a chatbot aimed to answer FAQs related to urban transportation –namely schedules or card loss. More specifically, this master's thesis is based on the development of the NLU module in a task-oriented DS for Mugi[2], the card used to travel around Gipuzkoa, Basque Country. As in any other field, there exist a series of limitations that need to be addressed thoroughly. One of them is the fact that the NLU needs a minimum amount of data in order to be trained. The more data it receives, the better the results are. In order to face this, a corpus was compiled. This corpus was specifically designed for this context, rather than using any existing resource in the scientific community. The corpus was obtained following collaborative method.

A set of researchers were asked to take part adding a series of possible user inquiries into a data gatherer we developed for this particular task. They were told to include utterances for the three domains we were working with, namely *in domain* (ID), *out-of-domain* (OOD) and *neutral* (N). Along with the instructions, they were showed a set of examples to the petition so that the participants would know what to do. In total, 597 input utterances were gathered. All participants either had Spanish as one of their mother tongues or had a native-like level.

All of this data was aimed to be expanded in order to form the largest corpus possible. Thus, the goal was to overcome the problem of data scarcity, so that any future research works would not be limited by the size of the training corpus. The following lines show a set of examples gathered.

1. Dónde puedo recargar la MUGI
   Where can I top-up my MUGI card

2. Dónde puedo usar la tarjeta
   Where can I use my card

3. La tarjeta tiene descuentos para jóvenes
   Does the card have discounts for the youth

4. Cuáles son las paradas de la línea 35
   What are the stops on the 35th line

The gathered corpus contained a total of 4634 unique tokens, which make a total of 597 utterances. After applying a POS tagger to the whole input corpus, the number of words per lexical category was obtained and gathered in Table (1). It shows that there is a predominant amount of nouns and verbs, along with a close amount of determiners. These constitute a basic linguistic structure, and the order of the constituents defines a language in typological terms. The tags used in Table (1) are specified a the SpaCy documentation online[3].

[2] https://www.mugi.eus/index.php/en/

[3] https://spacy.io/api/annotation

Lexical category	No.
NOUN	940
VERB	774
DET	762
PROPN	363
ADV	112
AUX (VERB)	346
SCONJ	143
CCONJ	53

Table 1: Lexical categories in the first corpus gathered

3.1 Preparing the Corpus for Expansion

There are two expansion or synonym extraction techniques described in Chapter 3. The first expansion technique involved tagging the input corpus. For this purpose, a Part of Speech (POS) tagger was used, via SpaCy (Honnibal and Montani, 2017). This annotation step is crucial when wanting to perform any machine learning task over a text-collection. However, as in any other NLP task, there is a major drawback: not all the entities might be tagged. This is something expected and the reasons behind those lost annotations can be the tagger itself not being able to recognize the entity or some of them not being quite correct in the language, e.g *xxxxx* or *Mugiiii*. In addition to these two case scenarios, there was a third group of inputs in this work that had an impact on every task applied —some inputs were written in two languages, English and Basque (14). This mixture is caused mainly by the diglossia existing in the territory in which the NLU from Chapter 1 is developed. Sociolinguistics define diglossia as a situation in which two varieties of the same language or two languages are used inside a single language community. This connection derives in a group of bilingual speakers that often use one the languages to designate a concept in the other language. This is exactly what happens within the corpus. There are a set of toponyms that have been written in Basque, such as *Leintz Gatzaga*, as well as words like *txartela*, which is translated as 'card' in English.

Once the corpus was tagged, there was another issue identified after applying the experiments explained in Chapter 3. As shown in Table (1), there were some entities such as determiners that appeared in the text almost as much as verbs or nouns did. These entities, along with others, do not have much semantic weight for user queries. The experiments were going to be carried out using the whole input corpus, but regarding the expanded results, a filtering was considered. This filter would keep those counterparts that corresponded to nouns and verbs, and therefore the final expanded corpus was expected to contain the same utterances from the beginning, with different versions for the selected lexical categories, as in a fill-in-the-gaps task. For instance, for the input where it says *La tarjeta tiene descuentos para jóvenes*, the hypothetical expanded version could be *La tarjeta tiene ofertas para chavales.*

The second expansion technique required another set of processes. As explained in

Chapter 3, this time the corpus needed a Word Sense Disambiguation (WSD) analysis for each term. For this purpose a series of NLP utilities provided by the IXA pipeline (Agerri et al., 2014) were used. They were namely POS tagging, tokenizing, lemmatizing and WSD steps. All of these were necessary in order to obtain the final required analysis, which are words disambiguated at sense level. Once the words were disambiguated, they were assigned with an Interlingual Index identifier (ILI) and a confidence score that rated how probable it was for that sense to be the right one. ILI identifiers correspond to the synsets in WordNet, a lexical database for English and other languages Miller (1998). The Spanish version was used for the synonym extraction, the one described in Gonzalez-Agirre et al. (2012). In this work, researchers describe the updated version of the Multilingual Central Repository (MCR) [4], developed in Atserias et al. (2004), which utilizes ILIs. These identifiers correspond to the 3.0 version of the database, and integrates WordNets from five different languages: English, Spanish, Catalan, Basque and Galician. For this Master's Thesis the Spanish version is used. By means of the ILIs, and after choosing the sense with the highest confidence score, a series of synonyms, hyponyms and hyperonyms were extracted. These will be evaluated in Chapter 3.

3.2 A Corpus for Out of Domain Detection

This section focuses on the procedures the corpus underwent after finishing the expansion techniques described in Chapter 3. After analyzing the results obtained with the expansion techniques, there were a set of entities that even though they belonged to the language, where considered to be out-of-domain. These OOD entities were obtaining a set of non-interesting counterparts, and therefore a series of scoring methods were developed to face this issue. Those methods needed the following preparation.

First, the new goal was set: carrying out a linear regression and a Support Vector Machine (SVM) using linear Kernel. These systems would be the ones in charge for the domain discrimination. For this purpose, the input corpus was ranked regarding the Term Frequency-Inverse Document Frequency (TF-IDF) score explained in Chapter 3. The resulting list of words gave an insight of the relevance of each of the terms along all the documents in the corpus—in this case, utterances. Then, there were three datasets prepared, with a tokenized and downsized version of the initial corpus, as we were using just nouns and verbs for this experimental phase. The corpus was labelled manually regarding the domain, namely OOD, N and ID. Next, there were two aspects to be taken into account with the corpus labelling. One of them was the fact that the variety of labels should be balanced, as well as their lexical category. The process of balancing up the corpus' features is called stratification, which in this case involved adding more OOD entities to the corpus, half nouns, half verbs. After the stratification, the statistics of the presence of each label remained as follows: 27.31% for N, 39.79% for ID and 32.88% for OOD, as well as 262 nouns and 245 verbs.

Once the datasets were suitable for the experiments, the words in the corpus needed

[4] https://adimen.si.ehu.es/cgi-bin/wei/public/wei.consult.perl

a feature vector to be represented. On of the one hand, there was the possibility of applying a typical vectorizer such as Count Vectorizer or TF-IDF Vectorizer, all provided by Scikit-Learn (Pedregosa et al., 2011). Instead, an array was created formed by a set of semi-redundant feature-vectors. These vectors were the result of a series of similarity operations described in Chapter 5 that involved a specific term selection and the calculation of cosine similarity between lists. The three-word set consisted of the following:

- Ten terms from the domain
- First thirty words in the TF-IDF ranking
- First, last and middle 20 words in the TF-IDF ranking.

It is rather important to point out that the reason behind the creation of this array is that semantic relationships between the words were captured, and therefore the results obtained would show whether these relations play an important role in a classifying task or not. By means of these selections three types of representations were obtained of the totality of the data: first, a small and very domain-dependant dataset, second, a wider in variety dataset but still containing elements belonging to very distinct classes and third, a final dataset containing samples of the three domains. The main goal of this task was that see whether there was a need to refine the word selection or, on the contrary, the specific way of creating the feature vectors did not make any difference.

Next chapter will describe and go through the first expansion techniques we carried out using the processed corpus described at the beginning of this section.

4 First Steps towards Expansion

In this chapter we will be presenting a set of methodologies that constitute first steps towards expansion techniques in NLP, along with a section for evaluation of the results obtained. We will start explaining two techniques we used for expansion trials, followed by a section describing the results obtained for evaluation purposes.

4.1 First Expansion Method: FastText

This section described a technique in which similar terms are obtained using word embeddings. The reason behind this lies on the nature of word embeddings, where semantically similar terms tend to obtain close vectorial representations in geometric space.

For this purpose, FastText's pretrained embeddings for Spanish (Grave et al., 2018) were used. These embeddings have been trained using Common Crawl (Crawl, 2019) and Wikipedia[5]. The pretrained model was downloaded using the FastText module by Gensim, thus obtaining a large amount of data suitable for machine learning purposes.

Simultaneously,the need to tag the corpus at syntactic and morphological level was identified. The retrieved expansion counterparts needed to be similar not only semantically, but also syntactically and morphologically. For this specific purpose the corpus was tagged using the Part of Speech (POS) tagger provided by SpaCy (Honnibal and Montani, 2017). Afterwards, a set of preprocessing techniques were applied over the corpus that included tokenizing, orthographic correction, removal of special characters and lowercasing. Lemmatization was purposely avoided, as the intention was to obtain expanded and conjugated or inflected counterparts. In that way, a final step of morphological reinflection will be skipped.

Next, in order to retrieve expansion candidates from the pretrained model, a probability of 0.2% was assigned to each of the words from the corpus and a maximum of 20 possible words that fell into that probability were set to be retrieved. By means of a embedding extraction functionality in the Gensim module, the words in the corpus obtained their embedding representation while finding semantically similar counterparts at the same time.

After a preliminary first inspection, it was observed that many of the constituents did not contribute much to the expanded corpus. These constituents were mainly adverbs, determiners and adjectives. A further filtering was applied by retrieving those constituents with the most semantic content— nouns and verbs. This filtered version of the expansion counterparts is meant to be applied to the expanded version of the corpus, where all the utterances will remain the same, and they will be repeated with the expanded synonyms in the gaps corresponding to nouns and verbs. For example, for the utterance *Dónde se carga el saldo* (where can I top) we will have *Dónde se abona el dinero*, up to as many times as counterparts there are.

[5] https://en.wikipedia.org/wiki/Main_Page

4.2 Second Expansion Method: WordNet

In this section WordNet (Miller, 1998) is used as source ontology for another expansion trial. The process started with a preprocessed corpus, with no syntactic or lexical information annotated. For this specific purpose, the IXA-pipeline was used (Agerri et al., 2014). Currently, the IXA-pipeline provides the following linguistic annotations: sentence segmentation, tokenization, Part of Speech (POS) tagging, lemmatization, Named Entity Recognition and Classification (NERC), constituent parsing and co-reference resolution. These tools are meant to ease any type of research work in NLP. In this thesis tokenization, POS tagging and lemmatization modules were used. All of these result in a NAF format document, which was used for the following step: Word Sense Disambiguation. Along with the IXA pipeline, there is a third-party tool named NAF-UKB (Agirre and Soroa, 2009), which is a graph-based WSD method that gives us some valuable information: a set of word IDs that correspond to their synset representation in WordNet. Graph-based WSD methods are particularly suited to disambiguate word sequences (Agirre and Soroa, 2009).

Once the NAF-UKB tool was applied, the words were disambiguated and were assigned with a confidence score for each of the senses assigned. At this point, the term associated to the highest confidence score would be used for the synonym extraction. For this purpose, a work [6] developed by Agirrezabal et al. (2019) was used. The adaptated code for this Master's Thesis extracts the information belonging to the level desired —in my case, the word form (wf) section given by the NAF-UKB tool where the ID is stored— and uses the NLTK (Loper and Bird, 2002) module for WordNet. An example of the synonyms obtained is shown below these lines.

5. Quería
   Necesitar, querer, precisar, necesitar
   (I/he/she/we) wanted

6. Subió
   Subir, levantar, elevar, alzar
   (He/she/it) went up

4.3 Evaluation

This section is divided in two parts. First one will go through a qualitative evaluation of the results, with more exhaustive examples of the expansion counterparts obtained and second one will present a quantitative evaluation with some metrics.

4.3.1 Qualitative Evaluation

This quality estimation will be divided in two more subsections, where the first one will go through the results obtained via FastText and the second one will examine those from

[6] `https://github.com/dss2016eu/codefest/tree/master/nlp_lac`

WordNet.

FastText In this context, FastText is considered a data-driven expansion technique. These counterparts are of a different nature (7) than those obtained with WordNet.

7. Cuánto cuesta hacerse la tarjeta
   Cuánto costó hacerse una **tarjetita**
   Cuánto cuesta hacerse una **terjeta**
   How much does it cost to obtain a card

8. Es personal e intransferible
   Es **perosnal** e **instranferible**
   Is it personal and nontransferable

9. Cuál es el porcentaje de chóferes mujeres en DBUS
   Cuál es el **pocentaje** de chóferes **muejres** en DBUS
   Cuál es el **procentaje** de chóferes féminas en DBUS
   What is the female driver percentage

10. Puedo recuperar el dinero
    Puedo recuperar el **dinerillo**
    Can I get back my money

11. Quiero saber mi saldo
    Queiro saber mi saldo
    I want to check my credit

12. Se me ha roto mi tarjeta Mugi
    Se me ha **descosío** mi tarjeta Mugi
    My Mugi card is damaged

13. Cuánto vale un billete a Amara
    Cuánto vale un **billetico** a San Agustín
    How much does it cost a ticket to Amara

Words in bold represent those counterparts that even though might be orthographically incorrect, or might not be found in a dictionary, they represent real word inputs. Retrieving these should be regarded as an important step towards training an intelligent NLU module, a module that will be able to understand both linguistically pure and correct inputs and pragmatically valid inputs, regardless of their orthography or colloquial register.

Similarly, some of the similar terms did not belong to the domain. Due to the orthographic errors found in the initial corpus, a series of preprocessing steps performed to remove any entity that did not belong to the working language of this thesis, but those that were part of the language remained untouched. For instance, there were with a pair of entities that were just 1 deletion apart (Levenshtein distance): *saldo* - *sado*. *Saldo* (credit) was part of the original vocabulary, and many user inputs that were gathered included it, except for one concrete case, where a typographical error in the word *sado* resulted in a series of counterparts that were not of our interest. This is the case for examples in (14-17).

14. Cuándo caduca el **sado**
    Cuándo caduca el **bdsm**
    When does the sado/bdsm expire

15. Cómo se va a **Bilbo**
    Cómo se va a **Frodo/Gandalf**
    How do I get to Bilbo/Frodo/Gandalf

16. Cómo cargo la tarjeta en **Bilbo**
    Cómo cargo la tarjeta en **Bolsón**
    How do I top-up my card in Baggins

17. Es posible su uso fuera de Dinastia
    Es posible su usos fuera de **dinastias**
    May I use it out of the dynasty/dynasties

First, the word *sado* is displayed. In this context, it is meant to be *saldo*, which is the Spanish word for credit. Due to a typographical error the resulting word in the input came out to be *sado*, which is also a contraction for sadomasochism. Therefore the similar words that came out were never words related to credit or money, but to the sexual practice.

In the case of *Bilbo*, in the context of the transportation domain is supposed to be the Basque word for Bilbao, a city from the territory. Similarly, there exists the word *Bilbo* in Spanish, but it is used to designate the character from the J.R.R Tolkien saga, and thus, the pretrained embeddings for Spanish provided us with the set of words that were found to be more similar to the input word, and these are other character names from the saga such as Gandalf or Frodo, or even the last name of the character Bilbo, Bolsón (Baggins).

Last term listed is *Dinastía.* This is another case of bilingualism playing a role in what the retrieval process. Donostia is the Basque name for San Sebastián, another city from the territory and the one in which the MUGI card is used. On the contrary, if you introduce a typographical error and write down *dinastia* what you obtain is the Spanish word for dynasty. Thus, the synonym retrieved did not correspond to other city names. Instead, the plural form for dynasty was obtained, *dinastias.*

Those are OOD entities, and handling them is considered essential to ensure robust performance of embeddings on real-world tasks (Al-Rfou et al., 2013). This issue will be analyzed thoroughly in Chapter 5.

In addition to that, there was another aspect that needed consideration. Some expanded sentences only included a difference in casing regarding the original term and there were even leaked special characters in between words, such as the following:

18. Cuánto cuesta hacerse una tarjeta
    -Cuánto cuesta hacerse otra tarjeta
    How much does it cost to obtain another card

19. Se me perdido la Mugiiii
    Se me perdido la **CasablancaTaichi**
    Se me perdido la **ZXNjcmliaXJub3MgYSA8YSBocmVmPSJ**
    I lost my Mugi card*

20. Cómo sé cuál es el descuento
    Cómo sé qué es el **descuento.-**
    How do I know my discount

21. Hasta cuándo podré utilizar el descuento de joven
    Hasta cuándo podré utilizar el **descuento50** de joven
    Hasta cuándo podré utilizar el **descuento.-** de joven
    What time does my youth discount finish

22. Se pondrán servicios especiales durante Semana Grande
    Se pondrán **deservicios** especiales durante **-Semana Grande**
    Will there be special services during the Grand Week

In this context, the expansion results are looked at explicitly, but this is not the case if the expansion is being carried out in a real case scenario. Knowing this, the presence of these non-valid entities need to be addressed using an automatized technique. However, some of them are derived from correct inputs, such as *cuánto* or *descuento*. In order to avoid these undesired entities to the extent possible, there are some solutions to be considered for the future, such as a change in the similarity probability assigned at the beginning or the reduction of the number of counterparts desired.

At the same time, many counterparts were considered suitable for expansion purposes, despite all the noise or repetitive entities:

23. La tarjeta tiene descuentos para jóvenes
    La tarjeta **tendrá ofertones** para **jóvenes**
    La tarjeta **tiene ofertazos** para **jóvenes**
    La tarjeta **tendrá cupones** para **jóvenes**
    Su tarjeta **tiene descuentos** para **adolescentes**
    La tarjeta **tiene descuentos** para **chicos**
    La tarjeta **tiene promociones** para **jóvenes** *Does the card have discounts for the youth*

24. No te olvides la tarjeta MUGI si vas en autobús
    No te olvides la tarjeta MUGI si vas en **microbús**
    No te olvides la tarjeta MUGI si vas en **tren**
    Do not forget your MUGI card if you use the bus

25. Se me roto la txartela
    Se me **quebrado** la txartela
    My card is damaged

26. Dónde puedo mirar mi saldo
    Dónde puedo mirar mi **balance**
    Where can I check my balance

27. Cuál es el porcentaje de chóferes mujeres en DBUS
    Cuál es el porcentaje de chóferes **féminas** en DBUS
    What is the female driver percentage in DBUS

28. Puedo recuperar el dinero
    Puedo **restituir** el dinero
    Puedo **recobrar** el dinero
    Puedo **recuperar** el dinero
    Puedo **restaurar** el dinero
    May I refund my money

Those terms in bold represent nouns and verbs that have been successfully expanded with a set of synonyms regarded as appropriate for the extended version of the corpus. These include some transports such as *autobús* with *microbús* and *tren* (train), or the set of words derived from *descuentos* (discounts) that even though they range in style from more formal to colloquial they are equally valid.

WordNet This section explores another expansion technique using, this time, a lexical database, namely WordNet and a set of word candidates that were used for synonym extraction purposes.

Original term	**Expanded terms**	**Translation**
Tipos	Tipo, clase, tipos	*Types*
Error	Falta, error, fallo	*Error*
Mujeres	Hembra adulta, mujer	*Women*
Condiciones	Circunstancias, condición, situación	*Conditions*
Pierde	Prescindir, renunciar, perder	*To lose*
Cobran	Cargar, cobrar	*To charge*
Crear	Causar, realizar, hacer, crear	*To create*
Repararlo	Atender, arreglar, reparar	*To repare*

Table 2: Set of correct counterparts extracted from WordNet

Table (2) shows that all of them contain the original word amongst the expanded terms, lemmatized, as in *repararlo* there is *reparar*, and sometimes it is also possible to find the original noun, in different numbers, as in *tipos* having *tipo* and *tipos* as possible answers. This might be regarded as a further step to take in processing if the results are implemented.

There are some other cases where the original terms obtained more counterparts than others, as shown in Table (3).

Original term	**Expanded terms**	**Translation**
Personas	Ser humano, persona, mortal, individuo, humano, alma, alguno, alguien	*People*
Conseguir	Tomar, sacar, conseguir, obtener, adquirir	*To achieve*
Ir	Salir, partir, marchar, dejar, ir	*To leave*
Olvida	Tirar, expulsar, olvidar, echar, deshacerse, desechar, descartar, arrojar	*To forget*
Lleva	Sostener, soportar, poseer, portar, llevar, contener	*To bring*
Función	Poder, jurisdicción, función, competencia, atribución	*Function*
Niños	Muchachuelo, nene, chiquillo, chico, chaval, alevín, criatura, crío, jovenzuelo, mozalbete, menor de edad	*Children*
Ocurre	Acaecer, acontecer, pasar, llegar a ocurrir, ocurrir llegar a pasar, pasar, sobrevenir, tener lugar, suceder	*To happen*

Table 3: Long lists of counterparts that can be obtained

It should be noted that some of these lists in Table (3) include a set of terms that are only suitable within the context. For instance, for *olvida* there are, amongst others, *deshacerse* and *desechar*. These two, out of context, represent synonyms of the word in Spanish for *to get rid of*, whilst in a different context can be regarded as *to forget*, in the sense of forgetting your card at home, for instance. These kind of context-dependant entities might not be taken into account by the system, specially since words are used to retrieve similar entities.

There are some other cases where regardless of the context the resulting entities have nothing to do with the domain. This is the case of some terms that in the disambiguation process have been assigned with a synset that is not suitable for our purposes. These non-valid entities are still part of the language, but the sense with which they were disambiguated was not the desired one. Again, these words need to be understood in context to get the right sense each time. If the model is fed with tokens, the result obtained corresponds to the most probable answer, which sometimes might not be aligned with the desired answer. This is the case for examples in Table (4).

These counterparts do not necessarily represent wrong expansion options. Even some of the options, such as *detectar*, *reconocer* or *núcleo* might be regarded as correct in our target context if were talking about a card or asking how to get to the center of the city, for instance. However, the rest of the counterparts represent synonyms of a sense of the word that do not belong to the context of the urban transportation, and thus will be considered invalid as in *cumplir lo prometido* or *arreglárselas*.

However, there is also a set of words that have not been successfully expanded. On the one hand, there are a set of entities that were not expanded with different entities but with themselves. Id est, they did not obtain any synonym. This is the case for words such as *imprimir* (to print), *llamar* (to call), *movilidad* (mobility) or *pasajeros* (passengers).

On the other hand, there are a set of terms belonging to the original corpus that did

Original term	**Expanded terms**	**Translation**
Salen	Salir, viajar, ir, arreglárselas, andar	*To exit*
Centro	Sustancia, núcleo, meollo, esencia, enjundia, corazón, centro	*Center*
Reconoce	Sentir, distinguir, discernir, detectar, reconocer	*To recognise*
Consigue	Cumplir lo prometido, salir adelante, tener éxito, triumfar, triunfar	*To achieve*
Puedo	Dar aviso, poder, echar, finiquitar despedir, despachar, dejar cesante	*To be able*
Terminales	Último, final, terminal	*Terminal*
Parada	Simplón, parado, lentorro,torpe	*Bus stop*
Gracias	Broma, ocurrencia, chiste, gracia	*Thank you*

Table 4: Set of incorrect expanded counterparts with WordNet

not obtain any type of counterpart, mainly because they did not have an ILI assigned in the beginning. That is to mean, in the disambiguation process, a set of ILIs were not obtained, resulting in a lack of counterparts for some of the inputs. Amongst these not valid terms there are terms such as *domingo* (sunday), *bici* (bike), *trámite* (procedure) or *txartela* (card). This is the case for *txartela*, which is the Basque term for 'card', and within a diglossic linguistic context it is rather difficult to predict which term a speaker will go for. Sometimes they will use toponymya in a certain language, and scientific terminology in the other. This is why we expect to have a certain loss in our results, in this case, a lack of expansion counterparts.

4.3.2 Quantitative Evaluation

This part of the evaluation is divided in three different phases. First phase will describe the criteria followed to consider a synonym invalid or valid. Second phase will be about the evaluation metrics used and third and last phase will provide an interpretation of the scores obtained.

The discrimination of a counterpart varies depending on the technique we used to retrieve counterparts. For both techniques, the possible cases where a retrieved term will not be kept are the ones listed in Table (5).

Next, the evaluation measures will be explained. The goal was to test whether the model had been able to retrieve a good percentage of counterparts out of the initial set of words, and for this purpose a sample of various words was taken out of the corpus. This sample contained up to 20 input terms from the corpus, along with their retrieved counterparts and they were evaluated following the criteria from Table (5). Whenever a counterpart met one of the problems, it was considered invalid. After validating the terms in the sample, accuracy was used as an evaluation metric:

The retrieved term...	**FastText**	**WordNet**
Only includes a difference in casing	✓	
Belongs to a different lexical category from the input word	✓	✓
It is the same word as the one in the input	✓	✓
Has no similar term retrieved		✓
Some ILI identifiers represent a synset in a sense we do not need		✓

Table 5: Term discrimination criteria for both expansion trials

$$Accuracy = \frac{Number\ of\ valid\ retrievals}{Total\ number\ of\ retrievals\ made}$$

By means of this metric, it was possible to obtain the number of results among the total number of cases examined. The selected list of words along with their similar terms retrieved can be observed in Tables (16, 17) from the appendix.

The evaluation of the number of counterparts will be carried out of three for each, as there is an unbalanced number of synonyms for every word. Considering three synonyms for each of the 20, the total candidates to be evaluated upon goes up to a total of 60. Out of these 60, FastText obtained a total score of 39 relevant synonyms whereas WordNet obtained a total of 21. After taking out the proportional score, FastText obtains a total of a 65% of accuracy and WordNet a total of 35%.

Results show that FastText is more accurate finding relevant counterparts. Taking into account the data used to train the embeddings for FastText, this is something expected to happen. Using word embeddings to find semantically similar terms has proved to be effective, and it proves one of our initial hypothesis. However, the linguistic aspects that differentiate one from another take an important role in their performance finding certain types of relevant synonyms. Using FastText, it is assumed that there are going to be some special characters, typographical errors and maybe even semantic noise, but the variety obtained represents a possible input in a real case scenario, while WordNet provides linguistic information that it is known for a fact that will be clean and strictly attached to the sense of the word. In addition to that, FastText was able to obtain results out of non-lemmatized elements, whereas all the process that followed to extract synonyms out of WordNet involved a necessary lemmatization step. Thus, FastText gives the ability to obtain an inflected form out of the input word, as in *funciona*, *funcionará* and *funcionó*, whereas WordNet inplies a final morphological generation step. This will be explained in further detail in the Chapter 6.

There are a couple of final observations that can be made out of the results. First and foremost, FastText's embeddings contained a considerable amount of typography errors, special characters, different casings even within the same word and entries in languages other than Spanish, despite the fact that each of the embedding sets are trained for a specific language. All of these could be translated as noise in our results, which can be both beneficial and problematic:

On the one hand, if the goal is to expand the corpus, it would also be suitable to retrieve a term with differences in typography. For instance, in cases where *tarjeta* obtained counterparts such as *targeta* or *tarjetita*. They are both variants of the same word, but they are not necessarily orthographically correct. Differences in spelling might be a loyal representation of what a user input looks like, and this is something to be taken into account when evaluating the expansion counterparts. On the other hand, WordNet represents a lexical database that contains not only synonyms but also manages to interlink a certain term with its hyperonyms or hyponyms. If the expansion methodology developed is able to extract every related term, including hyponyms and hyperonyms, the results obtained will add linguistic coverage. For instance, if the user finds it difficult to recall the word *crear* for a query where the user is asking the procedure needed to obtain a personal card, and the researchers utilized a resource that works similar to WordNet, they can also try with terms such as *hacer* or *realizar*.

5 Out-of-Domain Detection

During the orthographic correction phase, there was a new issue that arose— OOD entities that were orthographically correct. These types of entities are not suitable for expansion, as they obtain counterparts or synonyms that add noise to the expanded data. In this case, OOD terms are often derived from a typographic error, or a bilingual input that includes terms written down in a different language, as seen in example 14. To avoid including noise in the expanded corpus, a method or function f to filter out these OOD terms needs to be designed. In order to re-use the word-vector models that are being employed in the utterance expansion, two different methods are presented in this section, both relying on the Cosine similarity, measure that has been widely used to identify similar terms and utterances in the literature (Sidorov et al., 2014) (Huang, 2008), (Dehak et al., 2010), (Tata and Patel, 2007). Both methods have the same base hypothesis, that same-domain terms will have a higher similarity than terms of different domains. Under this assumption, the cosine similarity scores are suitable features to detect ID/OOD terms before performing the utterance expansion, as well as avoiding expansion candidates which are not related with the target domain.

5.1 Sampling the Reference terms

As the cosine similarity requires two vectors to compute the score, to detect if some term t_i is ID or OOD, a set of reference terms $t_j \in \mathcal{T}_{ref}$ is required. This set of reference terms $\mathcal{T}_{ref}$ needs to be selected beforehand and knowing if these terms $t_j \in \mathcal{T}_{ref}$ are also ID, OOD and N.

5.1.1 Frequency and Similarity Measuring

The main goal of this section is sampling the group of words that will be used to design the domain-discrimination scoring method. For instance, regarding the transportation domain, having words like *bus* or *tarjeta*, the model is meant to determine that other words such as *saldo* or *línea* are domain-specific, because they belong to the same semantic field, whereas *dinastía* or *sado* are not.

The main idea is that each input word is assigned with a similarity score with respect to a set of terms considered to be relevant for the domain. Cosine similarity is explored as similarity score and Term Frequency-Inverse Document Frequency (TF-IDF) is used Qaiser and Ali (2018) to derive the set of terms relevant for the domain. Cosine similarity should help telling whether two terms are semantically distant or not. TF-IDF is a score that shows the relevance of keywords to some specific documents[7]. The score is higher when the term occurs many times within a small number of documents and lower when it occurs fewer times in a document, or occurs in many documents.

As a first step, TF-IDF was applied to the corpus and its terms were ranked according to their relevance. Then, two sets of words were selected. The first set, referred to as

[7] In this experimental context, documents are sentences.

Reference List, was considered part of the domain. The second set, referred to as Candidate List, contained ten words from the domain and ten words from out of the domain. The selection for the Reference List was carried out manually, out of the first thirty words in the ranking, whilst the rest of the terms in the Candidate list were sampled as valid candidates by FastText when performing the expansion described in Chapter 3.

	ID	**OOD**
Reference List	Tarjeta, pagar, recargar, perdido, bus, descuento, foto, caducar, roto, saldo	
Candidate List	Autobús, línea, funcionar, bono, anónima, tren, joven, familia, abonar	Numerosa, pegar, sado, rito, dinastía, Frodo, gamusino, lagarto, heterogénea, feto

Table 6: Set of selected words for the initial scoring method

As Table (6) shows, *sado* (sadomasochism), *feto* (fetus) and *dinastía* (dynasty) are those cases where the orthographic distance plays an essential role in the meaning of these inputs, whilst *Gandalf*, *lagarto* (lizard) and *Frodo* are derived from other type of terms. *Gandalf* and *Frodo* come from a mixture of bilingualism in terminology and context, as explained in Chapter 2. *Bilbo* represents two different concepts: one of them, the Basque name for a city, and the other one the character from a book. On the contrary, with terms such as *lagarto* the situation is different. There are some terms that even though belong to the same language, they might be too dependant on the context. This is the case for words such as *topo*, which is the name of an urban transportation used in Donostia-San Sebastián and at the same time represents the Spanish word for mole, the animal.

As stated before, both lists were then compared using cosine similarity. The main hypothesis was that semantically similar words would obtain higher scores, and the scorer would be able to tell what is part of the domain. By means of a specific module and a set of pretrained embeddings described in Chapter 2, all the words in the lists obtained a representation in embedding space. These embeddings were compared and the results obtained are shown in (7, 8, 9, 10).

As Tables (7, 8, 9, 10) show, scores between semantically similar terms tend to range between 0.2 and 0.4, whilst semantically distant terms shown in Table (8) may be closer to 0.1, overall. There exists a tendency to obtain a higher score between those words that may occur within the same context.

In order to calculate a threshold at which terms would be discriminated, three different metrics were evaluated:

29. Taking out the mean value of all the values obtained

30. Taking out the mean value for the highest three values

Candidate/Reference	**Tarjeta**	**Pagar**	**Recargar**	**Perdido**	**Bus**
Autobús	0.2418714	0.20925952	0.17679633	0.13575219	0.7447229
Línea	0.3233888	0.1656909	0.21093379	0.17414205	0.2381076
Funcionar	0.04165129	0.30037066	0.36398408	0.15214044	0.2587137
Bono	0.3624298	0.37069902	0.25885865	0.16913082	0.2587137
Anónima	0.16803275	0.15721227	0.04623211	0.10856031	-0.00222818
Tren	0.15637097	0.11652669	0.14988649	0.18362099	0.63578665
Joven	0.15019077	0.10485408	0.04648217	0.26704416	0.15479283
Familia	0.23092997	0.14713691	0.05839755	0.19742098	0.0757714
Abonar	0.32533032	0.7054469	0.41488564	0.19140862	0.13801192
Numerosa	0.24031907	0.14133096	0.10481069	0.13822204	0.02608366

Table 7: Cosine similarity scores regarding the ID Candidate List

Candidate/Reference	**Descuento**	**Foto**	**Caducar**	**Roto**	**Saldo**
Autobús	0.18294775	0.20147337	0.0461246	0.11952911	0.18143244
Línea	0.1778488	0.29070395	0.11341325	0.16313565	0.18635169
Funcionar	0.09538276	0.10031297	0.35813916	0.28751892	0.13713077
Bono	0.49779898	0.08504781	0.23674789	0.08831848	0.4286416
Anónima	0.15083347	0.19052164	0.08256797	0.04146213	0.11005142
Tren	0.08543938	0.19417073	0.04380835	0.14417888	0.16900735
Joven	0.14098422	0.20586029	0.03300728	0.19975251	0.15825826
Familia	0.02871594	0.24451499	-0.00266129	0.12480344	0.0388367
Abonar	0.32507178	0.11822005	0.26531035	0.09046825	0.37793818
Numerosa	0.1423008	0.20600334	-0.00176542	-2.6237198e-05	0.15900846

Table 8: Cosine similarity scores regarding the ID Candidate List II

Candidate/Reference	**Tarjeta**	**Pagar**	**Recargar**	**Perdido**	**Bus**
Pegar	0.1741307	0.30015492	0.30925208	0.15564229	0.09348263
Sado	0.04165129	0.12131473	-0.00878675	0.02312869	0.07147245
Rito	0.07705189	0.13781375	0.04037283	0.14743191	0.16022694
Sueldo	0.18494105	0.4462078	0.09771721	0.17403708	0.1826075
Dinastía	0.10037293	0.02200596	0.00463529	0.06721257	-0.0583726
Frodo	-0.00727617	-0.00525666	0.01191583	0.14925757	0.14756462
Gamusino	0.08023829	0.06332164	0.03117045	0.15202478	0.09558433
Lagarto	0.06158754	0.10730966	0.01975991	0.19221307	0.17513919
Heterogénea	0.12583604	-0.02666428	0.02129127	0.06599858	0.07695718
Feto	0.0820426	0.06992966	0.01016528	0.10113156	0.12154278

Table 9: Cosine similarity scores regarding the OOD Candidate List I

Candidate/Reference	**Descuento**	**Foto**	**Caducar**	**Roto**	**Saldo**
Pegar	0.18706866	0.24116945	0.21462588	0.2961324	0.05661663
Sado	0.0597038	0.05456895	0.04991329	0.08657002	0.07388252
Rito	0.13622165	0.08485822	0.09617086	0.17559357	0.1621922
Sueldo	0.29570138	0.13662682	0.07699826	0.05380964	0.31516558
Dinastía	0.01267642	0.08652832	-0.00378286	0.0727656	0.05385095
Frodo	-0.03822013	0.17094837	-0.0058336	0.20086414	0.03073874
Gamusino	0.04801809	0.13210028	0.09436445	0.12157971	-0.00519646
Lagarto	0.12025395	0.17003082	0.00099259	0.172036	0.09353891
Heterogénea	0.01657149	0.1624482	0.01999669	0.07052121	0.07493159
Feto	0.04043418	0.08034457	0.09436002	0.08433678	0.14075986

Table 10: Cosine similarity scores regarding the OOD Candidate List II

	Candidate List/Mean Values	**All**	**Highest 3**	**Highest and Lowest**
In	Autobús	0.22399096	0.39861795	0.39542374
	Línea	0.20437165	0.28406676	0.21840101
	Funcionar	0.27045282	0.340831309	0.22968341
	Bono	0.27563	0.43237987	0.29142338
	Anónima	0.09789	0.17192222	0.09414672
	Tren	0.18787	0.33785948	0.3397974
	Joven	0.14612	0.22421897	0.15002572
	Familia	0.11438	0.22428865	0.120926841
	Abonar	0.2952	0.49942353	0.39795756
Out	Numerosa	-0.1682	0.20177696	0.1192768
	Sado	0.05677	0.09392241	0.0562639
	Pegar	0.20282	0.30184647	0.18293435
	Rito	0.12179	0.16600424	0.10798320
	Dinastía	0.03578	0.08655560	0.02100016
	Frodo	0.06547	0.17369003	0.08132199
	Sueldo	0.19638	0.35235825	0.25000870
	Lagarto	0.11128	0.17979608	0.09660282
	Heterogénea	0.06078	0.12174713	0.06789196
	Feto	0.09165	0.12114473	0.07546256
	Gamusino	0.08632	0.13523492	0.07341416

Table 11: Mean similarity scores for each of the terms shown in (7, 8, 9, 10)

31. Calculating the mean value of the highest and the lowest score for each term.

The mean values obtained via all the operations are shown in (11). The first two metrics, in (11) referred to as 'All' and 'Highest 3' obtained a set of values that did not enhance terms belonging to the domain. Overall, all the terms obtained scores that were not very apart. On the contrary, 'Highest and Lowest' showed a difference that favoured ID terms over the OOD ones. This last metric was chosen to determine the threshold in the following experiments.

The results above prove that TF-IDF is a suitable technique to determine which terms are relevant to the domain. Additionally, it is also proven that using a cosine similarity, combined with calculating the mean value of the highest and lowest scores result in a scorer that is able to discriminate between ID, OOD and N. As a final step, a threshold shall be established at which terms would be discriminated. Looking at the scores obtained (11), terms could be classified at three different points: 0.1 for 'All', 0.22 for 'Highest 3' and 0.2 for 'Highest and Lowest'.

5.2 Regression and Classification for Domain Discrimination

As Section 5.1 demonstrates, TF-IDF combined with cosine similarity can be used to discriminate ID and OOD terms, yet, these preliminary experiments involve manual work in the selection of the Reference List. This section focuses on developing an automatic scoring method that will be used for the domain-discrimination task. For this purpose, two different classifiers are introduced: a linear regression system and a Support Vector Machine (SVM) using linear Kernel.

A linear regression algorithm is composed by two data features, a dependent variable real-value y and an independent variable value x. By means of this algorithm, y is predicted based on x, as the model finds a linear relationship between the input, namely x and the output, y. In this case, the input consists of a n-dimensional vector matrix composed of similarities between terms (the cosine similarities between the candidate term $c_i \in \mathcal{C}$ and the candidate terms sampled using the TF-IDF $t_j \in \mathcal{T}$) and the regression values range in [-1, 1] and represent the labels assigned to each of the terms in the corpus, that is 0 for *N*, 1 for *ID* and -1 for *OOD*. The linear relationship that the model finds is represented in the form of a straight line and corresponds to the following equation:

$$y = mx + b$$

The components in this equation are divided as follows: b is the intercept, m is the slope of the line and x represents the input values. By means of this operation, the linear regression will give us the most optimal value for the intercept and the slope to determine y in terms of the values of x.

The regression algorithm is meant to fit multiple lines on the data (x) points and returns the line that results in the least error. In other words, the algorithm finds the most optimal coefficients, which are the output that determine the change that y experiences regarding the input values. In this case, the goal was to observe the weight of each of the features in x in a regression task for the domain-discrimination purpose.

In our case, as the features of x_j are determined by the cosine similarity between the reference term $t_j \in \mathcal{T}$ and the candidate term $c_i \in \mathcal{C}$:

$$y_i = b + m_1 \cdot cos(t_1, c_i) + m_2 \cdot cos(t_2, c_i) + \cdots + m_{|\mathcal{T}|} \cdot cos(t_{|\mathcal{T}|}, c_i) +$$

Then, y_i, the predicted score, is a real-value which represents the Neutral label if it is close to 0, OOD if it is close to -1, and ID if it is close to 1. The regression method, as it weights each feature $cos(t_j, c_i)$ is specially useful to interpret how each reference term t_j contributes to the decision making.

In addition with the regression system, a Support Vector Machine (SVM) with linear Kernel was used. The combination of both systems provide a robust and tested mechanism for decision making. Support Vector Machines (SVM) are meant to find a hyperplane in an n-dimensional space that best classifies data points. Again, the labels ID, OOD and N are used.

The aim of this experimental phase is to prove the impact of the way in which the Reference terms are sampled when modelling a domain-discrimination system. The three datasets contained all the nouns and verbs from the original corpus, along with a set of labels that determined the class they belonged to. In addition to this, each of the datasets used a different Reference List to extract the features in x. The first selection contained ten words belonging to the domain, the second version contained the first thirty words from the TF-IDF ranking list and the third set contained sixty words, which were the twenty first and last words in the raking and twenty more belonging to the middle. The first selection was manually done, whilst the second and third word-selections were automatically sampled for the TF-IDF ranking.

Afterwards, each dataset was divided into train and test sets, with a 80-20 partition each. Then both classifiers were trained, the Linear Regressor and the SVM provided by Scikit-Learn Pedregosa et al. (2011). Next section describes an evaluation of the results achieved in more detail.

5.3 Evaluation

As described in Section 5.2, three different linear regression systems, along with three SVMs with linear Kernel were trained. The combination of these two systems applied over three distinct datasets aimed to formalize a scoring method designed for the primary domain-discrimination task. The main hypothesis favoured the largest dataset, as it contained a wider TF-IDF term variety in its feature vectors. The more options there are, the better the model learns about the domain and therefore, the better it classifies.

In order to evaluate the linear regression classifier, two scores were calculated: the Mean Squared Error (MSE) and the Coefficient of determination. The MSE calculates the squared error of the expected value versus the obtained value in classification context. The lower it is, the better the model works. The coefficient of determination, also known as R^2 or R squared, calculates the strength of the linear relationship between two variables. In statistic terms, it measures how differences in one variable can be explained by the differences in a second variable when making a prediction. R^2 acquires values between 0 and 1, the closer it is to 1, the better the models work. The scores obtained for the three regression models and classifiers are shown in Table (12):

In the case of the MSE, the lowest score is the one obtained with the dataset containing the largest feature-vectors, which is the one called 'sixty'. This means the model containing

	Ten	**Thirty**	**Sixty**
R^2	0.2649600253214639	0.2921653026060125	0.343589407408022
MSE	0.548438130954976	0.5283210745147832	0.48977009542594735

Table 12: Metrics for the evaluation of the Linear Regression

that dataset has the lowest squared error in its predictions. Regarding the coefficient of determination, on the contrary, the highest score is the best working one. In this case, the closest to 1 is, again, the 'sixty' dataset. Regarding the rest of the scores,they obtain lower scores for the coefficient of determination and higher scores for the mean squared error, which can be read as worse results. It is also possible to observe this looking at another set of results obtained via the linear regression system —coefficients. Coefficients show whether there is a positive or negative correlation between the independent variable X and the dependent variable Y. In regressions where there are multiple regression values, as in this case, coefficients calculate how much the dependent variable is expected to increase when that independent variable increases by one, holding all the other independent variables constant. In this case, the independent variable X contains n-dimensional feature vectors, the ones obtained calculating similarities between two sets of terms and the dependent variable Y contains the regression values, which range in [-1, 1], -1 for OOD, 0 for N and 1 for ID.

In this context, coefficients are the way to determine the weight of each of the features —similarities between words— regarding the domain classification. For instance, if a coefficient gets a value of 0.77, it means that feature is more prone to be ID, whereas a negative tendency shows that a feature is more probable to be OOD. Therefore, if a value is placed at -3 in 2, that t_j reference term may be considered as very OOD. Moreover, other reference terms might obtain positive or neutral values regarding a c_i candidate term, but whenever the reference term is similar to that one with a coefficient of -3 assigned, it will be placed as OOD by the system.

In the case of the SVM, results were evaluated using a classification report shown in Tables (15, 14, 13). The report shows the main classification metrics: precision, recall and F_1. The metrics are calculated regarding true and false positives as well as true and false negatives. In the case of precision, it is defined as the ratio of true positives to the sum of true and false positives. In other words, it shows the ability of the classifier not to label an instance positive that is actually negative. Recall is defined as the ratio of true positives to the sum of true positives and false negatives, the ability of a classifier to find all the positive instances. Finally, the F_1 score, it is defined as a weighted harmonic mean of precision and recall, and it is used to compare classifier models.

The reports show virtually the same that can be observed with the results obtained with the regression system. The more domain-related information the feature vectors have, the better the model works. The 'sixty' dataset obtained an accuracy of 0.69, whilst 'ten' obtained a 0.53. Similarly, the 'thirty' dataset obtained an in between result with an accuracy of 0.65, but again it is closer to 'sixty'.

Both set of results prove that for an OOD detection task it is advisable to train the

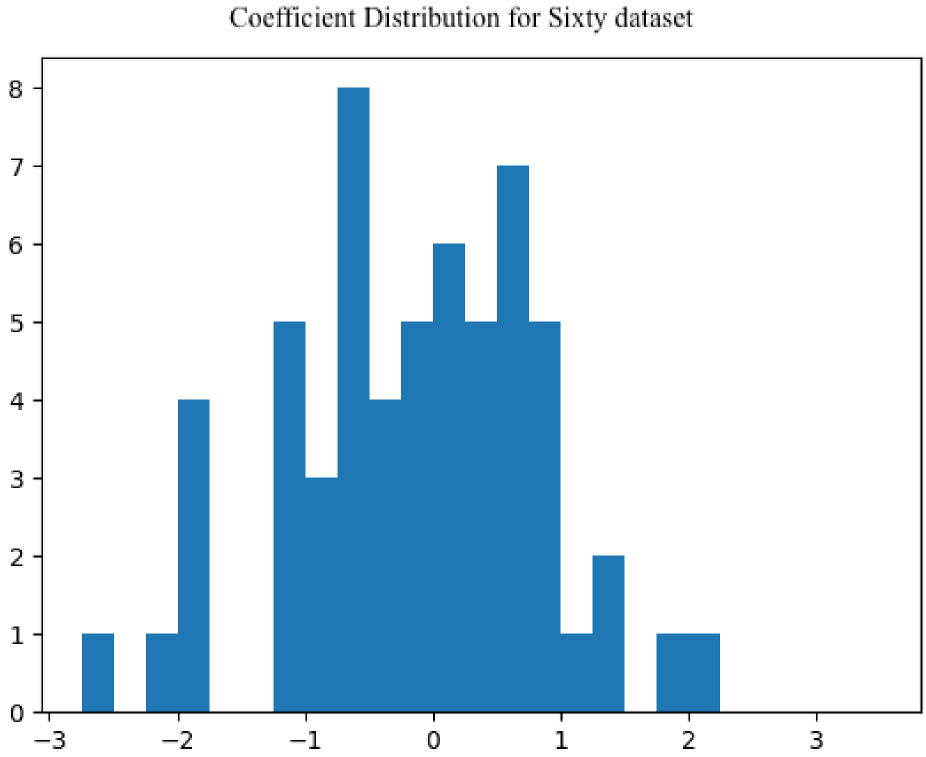

Figure 2: Coefficient distributional representation for ’Sixty’

	Precision	**Recall**	**f1-score**	**Support**
OOD	0.56	0.73	0.63	51
N	1.00	0.03	0.07	29
ID	0.48	0.64	0.55	39
Accuracy			0.53	119
Macro avg.	0.68	0.47	0.42	119
Weighted avg.	0.64	0.53	0.47	119

Table 13: Classification report for the ’ten’ dataset

models with terms belonging to the three domains —unlike Thirty and Ten datasets— selected out of a list ordered regarding the TF-IDF score for each of the terms. Additionally, the OOD detection task can be carried out automatically by means of Machine Learning techniques and TF-IDF weighting.

	Precision	**Recall**	**f1-score**	**Support**
OOD	0.66	0.76	0.71	51
N	0.52	0.48	0.50	29
ID	0.73	0.62	0.67	39
Accuracy			0.65	119
Macro avg.	0.64	0.62	0.63	119
Weighted avg.	0.65	0.65	0.64	119

Table 14: Classification report for the 'thirty' dataset

	Precision	**Recall**	**f1-score**	**Support**
OOD	0.75	0.80	0.77	51
N	0.52	0.55	0.53	29
ID	0.76	0.64	0.69	39
Accuracy			0.69	119
Macro avg.	0.67	0.67	0.67	119
Weighted avg.	0.69	0.69	0.69	119

Table 15: Classification report for the 'sixty' dataset

6 Conclusions and Future Work

This work is developed in the context of Dialogue Systems, more specifically within a project developing a chatbot for the urban transportation domain. The aim of the study was to develop a way to expand the data to train the NLU module in that chatbot. The main hypothesis was that the more data the module had, the better it would learn to process user inputs. First, the corpus was gathered and preprocessed. Once the corpus was clean, it had to be prepared in different ways to face two experimental phases.

Regarding the first experimental phase, the goal was to find ways to expand the gathered data. The first explored technique involved using word vectors for expansion. For this purpose, the corpus was tagged using Part of Speech tags and, by means of FastText's pretrained set of embeddings, the original terms were expanded with up to 20 counterparts each. The second explored technique used words from the original corpus to extract synonyms out of the WordNet lexical database. These words were tagged and processed with a WSD module that assigned an identification number to each of the words which were later used for synonym extraction. The results had to be filtered regarding a series of criteria, such as the expanded term had to belong to the same lexical category as that of the input and it also had to be coherent with the context (5).

Lastly, the second experimental phase involved a domain discrimination task, whose necessity was identified whilst analyzing the expansion results. For this discrimination task, the suitability of TF-IDF and cosine similarity as techniques to sample domain-relevant terms and discriminate between In-Domain (ID), Out-of-Domain (OOD) and Neutral(N) terms was first analyzed. Then, linear regression and SVM classification algorithms were explored for domain discrimination. Experiments were carried out on three different datasets, which involved tagging the initial corpus regarding the domain and three word selections out of the TF-IDF list. Throughout this work, a series of issues were identified that need some consideration. The approval of the expansion results depends on the nature of the resource used. Regarding FastText, the counterparts extracted were more loyal to the nature of a real input. In a real case scenario, it is expected to encounter inputs that contain special characters derived from mistyping, orthographic errors or maybe even diminutives. Looking at the results obtained with the pretrained embeddings, there are many entities that fit in these patterns. Thus, extracting expansion counterparts that are similar to the morphology of inputs might be helpful for the capability of the NLU. As for WordNet, the quality of the results is measured in a different way. All the counterparts obtained come from a lexical database, whose content meets three characteristics:

- All the elements found in the database are part of the language dictionary.
- All the elements are lemmatized.
- The lexical categories that can be found are adverbs, verbs, nouns and adjectives.

These three can be regarded as beneficial and counterproductive at the same time. Considering the linguistic side, the synonyms obtained via WordNet add a wider linguistic coverage. Instead of finding different spellings for the same entity, what the database provides is a set of synonyms that enhances the variety in the lexical aspect. These two techniques have proved to be effective at different levels, this is why the combination of both expansion trials could be seen as a way to maximize the advantageous aspects of both techniques.

However, the fact that in WordNet both the input data and the extracted data are lemmatized involves a further step of morphological reinflection. This reinflection would add the same conjugation or declination found in the original input and thus, the expanded utterances would be coherent. This last step can be analyzed as part of the future work, as for the time being there is no morphological generator that could be found for Spanish.

Additionally, there is the issue with the expansion counterparts obtained from the bilingual inputs. Some of them might be discarded by the scoring method designed in Chapter 5, but some might not. Taking into account the territory in which the chatbot is going to be used, it is probable that users do not distinguish between languages to designate certain concepts, such as *txartela* (card). Developing a bilingual chatbot would solve these kind of problems, therefore this might also be regarded as future work.

Finally, the results obtained with regard to OOD detection are promising. The initial analysis showed that it was possible to discriminate certain inputs using a combination of TF-IDF and cosine similarity, and so the regressor and the SVM were meant to perform basically the same task with bigger feature vectors. The results obtained with those two, as shown in Tables (15, 13, 14), have proved that it is possible to discriminate domains. However, the highest accuracy score is 0.69, which shows that there is still work to carry out, . Further experiments are needed for the domain-discrimination purpose, such as using more heterogeneous feature-vectors or combining other types of classifiers.

Additionally, there are other resources that may be taken into account for future expansion purposes, such as BabelNet as alternative database along with another morphology generator or other similarity metrics rather than cosine similarity that allow us to design effective scoring methods.

Overall, this work proves that it is possible to develop expansion techniques utilizing small samples of data. The implemented systems are capable of obtaining valid synonyms and have the ability to discriminate regarding the domain. These first results might open a door for future refining experiments that could look into applying other similarity metrics with additional databases for expansion or exploiting larger feature vectors for classification.

References

Rodrigo Agerri, Josu Bermudez, and German Rigau. Ixa pipeline: Efficient and ready to use multilingual nlp tools. In *LREC*, volume 2014, pages 3823–3828, 2014.

Eneko Agirre and Aitor Soroa. Personalizing pagerank for word sense disambiguation. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 33–41, 2009.

Manex Agirrezabal, Begoña Altuna, Lara Gil Vallejo, Josu Goikoetxea, and Itziar Gonzalez Dios. Creating vocabulary exercises through nlp. *Digital Humanities in the Nordic Countries. Proceedings, 2019*, 2019.

Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. Polyglot: Distributed word representations for multilingual nlp. *arXiv preprint arXiv:1307.1662*, 2013.

Jordi Atserias, Luıs Villarejo, German Rigau, Eneko Agirre, John Carroll, Bernardo Magnini, and Piek Vossen. The meaning multilingual central repository. In *2nd International Global Wordnet Conference, January 20-23, 2004: proceedings*, pages 23–30. Masaryk University, 2004.

Anja Belz. Automatic generation of weather forecast texts using comprehensive probabilistic generation-space models. *Natural Language Engineering*, 14(4):431, 2008.

Hongshen Chen, Xiaorui Liu, Dawei Yin, and Jiliang Tang. A survey on dialogue systems: Recent advances and new frontiers. *Acm Sigkdd Explorations Newsletter*, 19(2):25–35, 2017.

Francesco Colace, Massimo De Santo, Luca Greco, and Paolo Napoletano. Weighted word pairs for query expansion. *Information Processing & Management*, 51(1):179–193, 2015.

Common Crawl. Common crawl. *URl: http://http://commoncrawl. org*, 2019.

Najim Dehak, Reda Dehak, James R Glass, Douglas A Reynolds, Patrick Kenny, et al. Cosine similarity scoring without score normalization techniques. In *Odyssey*, page 15, 2010.

MeiXing Dong, Rada Mihalcea, and Dragomir Radev. Extending sparse text with induced domain-specific lexicons and embeddings: A case study on predicting donations. *Computer Speech & Language*, 59:157–168, 2020.

Aitor Gonzalez-Agirre, Egoitz Laparra, and German Rigau. Multilingual central repository version 3.0. In *LREC*, volume 2525, page 2529, 2012.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. Learning word vectors for 157 languages. *arXiv preprint arXiv:1802.06893*, 2018.

Matthew Honnibal and Ines Montani. spacy 2: Natural language understanding with bloom embeddings. *convolutional neural networks and incremental parsing*, 7(1), 2017.

Anna Huang. Similarity measures for text document clustering. In *Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008), Christchurch, New Zealand*, volume 4, pages 9–56, 2008.

Kristiina Jokinen. Learning dialogue systems. In *LREC 2000 Workshop: From Spoken Dialogue to Full Natural Interactive Dialogue-Theory, Empirical Analysis and Evaluation*, pages 13–17. Citeseer, 2000.

Sangkeun Jung. Semantic vector learning for natural language understanding. *Computer Speech & Language*, 56:130–145, 2019.

Edward Loper and Steven Bird. Nltk: the natural language toolkit. *arXiv preprint cs/0205028*, 2002.

Jonathan Mamou, Oren Pereg, Moshe Wasserblat, Alon Eirew, Yael Green, Shira Guskin, Peter Izsak, and Daniel Korat. Term set expansion based nlp architect by intel ai lab. *arXiv preprint arXiv:1808.08953*, 2018.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

George A Miller. *WordNet: An electronic lexical database.* MIT press, 1998.

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, 2010.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

Xuan-Hieu Phan, Le-Minh Nguyen, and Susumu Horiguchi. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *Proceedings of the 17th international conference on World Wide Web*, pages 91–100, 2008.

Shahzad Qaiser and Ramsha Ali. Text mining: use of tf-idf to examine the relevance of words to documents. *International Journal of Computer Applications*, 181(1):25–29, 2018.

Nirmal Roy. Word embedding models for query expansion in answer passage retrieval. Master's thesis, Delft University of Technology, 2019. `http://resolver.tudelft.nl/uuid:3152697f-eacc-42dc-a686-33948fd5ea3a`.

Grigori Sidorov, Alexander Gelbukh, Helena Gómez-Adorno, and David Pinto. Soft similarity and soft cosine measure: Similarity of features in vector space model. *Computación y Sistemas*, 18(3):491–504, 2014.

Min Song, Il-Yeol Song, Xiaohua Hu, and Robert B Allen. Integration of association rules and ontologies for semantic query expansion. *Data & Knowledge Engineering*, 63(1): 63–75, 2007.

Amanda Stent, Matthew Marge, and Mohit Singhai. Evaluating evaluation methods for generation in the presence of variation. In *international conference on intelligent text processing and computational linguistics*, pages 341–351. Springer, 2005.

Sandeep Tata and Jignesh M Patel. Estimating the selectivity of tf-idf based cosine similarity predicates. *ACM Sigmod Record*, 36(2):7–12, 2007.

Blaise Thomson and Steve Young. Bayesian update of dialogue state: A pomdp framework for spoken dialogue systems. *Computer Speech & Language*, 24(4):562–588, 2010.

Peng Wang, Bo Xu, Jiaming Xu, Guanhua Tian, Cheng-Lin Liu, and Hongwei Hao. Semantic expansion using word embedding clustering and convolutional neural network for improving short text classification. *Neurocomputing*, 174:806–814, 2016.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. *arXiv preprint arXiv:1508.01745*, 2015.

A Evaluation Tables for Expansion Results

Input words	**FastText**
Tarjeta	*Tajeta, tarjetita, terjeta, tarjeta*
Recargar	*Recargar*
Autobús	*Taxi, bus, autobús*
Puedo	*Debo, voy, puedo*
Saldo	*Insoluto, saldo, saldado*
Perdido	*Perdido, perdidos, recobrado*
Bus	*Bus*
Dinero	*Dinero, dinerillo, dineo*
Funciona	*Funciona, funcionó, funcionará*
Descuento	*Descuento, descuento.-, escuento, descuento50, oferta*
Roto	*Roto, descosió, rompido, rompe, descocido*
Viaje	*Viaje, viajecito, viajes*
Billete	*Billete, billetes, tiques, billetito, billetico*
Transbordo	*Transbordo, trasborda, transbordando*
Tren	*Autobús, tranvía, tren, autobús.-, microbús*
Cargar	*Cargar*
Ir	*Ir*
Metro	*Metro, metro.-*
Usar	*Usar, usarse, emplear, utilizar*
Quiero	*Quiero, necesito, pretendo, queiro, quise*

Table 16: Sample of similar terms obtained via FastText for evaluation

Input words	**WordNet**
Tarjeta	*Tarjeta, tarjetas*
Recargar	-
Autobús	-
Puedo	*Poder, finiquitar, echar, despedir, despachar, dejar cesante, dar aviso*
Saldo	-
Perdido	*Encuentro, recuperar, encontrar*
Bus	*Bus, embarrado*
Dinero	*Dinero*
Funciona	*Funcionar, operar*
Descuento	-
Roto	*Roto, estropeado*
Viaje	*Viaje, locomoción*
Billete	*Billete*
Transbordo	*Transbordo*
Tren	*Tren, ferrocarril*
Cargar	*Cargar, cobrar, picar, pinchar*
Ir	*Acudir, ir, proceder*
Metro	*Metro*
Usar	*Usar, utilizar, emplear*
Quiero	*Querer, requerir, precisar, necesitar*

Table 17: Sample of similar words obtained via WordNet for evaluation