CUKUROVA UNIVERSITY

INSTITUTE OF NATURAL AND APPLIED SCIENCES

PhD THESIS

Contextualized Intent Detection Using Generalized SemSpace and BLSTM

Elif Gülfidan TOSUN

Computer Engineering Department

September, 2023

CUKUROVA UNIVERSITY

INSTITUTE OF NATURAL AND APPLIED SCIENCES

PhD THESIS APPROVAL

Contextualized Intent Detection Using Generalized SemSpace and BLSTM

Elif Gülfidan TOSUN

Computer Engineering Department

This Doctorate Thesis was evaluated by the following Jury Members on 28/09/2023 and was approved by unanimity / majority of votes.

Jury : Prof. Dr. Umut ORHAN (Advisor) ……

: Prof. Dr. Selma Ayşe ÖZEL ……

: Assoc. Prof. Dr. Ali İNAN ……

: Asst. Prof. Dr. Mehmet SARIGÜL ……

: Asst. Prof. Dr. Çağatay Neftali TÜLÜ ……

This Thesis was written in the Department of Computer Engineering, Institute of Natural and Applied Sciences.

Thesis Number:

Prof. Dr. Sadık DİNÇER
Director
Institute of Natural and Applied Sciences

CONTENTS

CUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES

PhD THESIS

Contextualized Intent Detection Using Generalized SemSpace and BLSTM

Elif Gülfidan TOSUN

Advisor: Prof. Dr. Umut ORHAN

Department of Computer Engineering

ABSTRACT

In this thesis, a novel method called Contextualized Deep SemSpace has been developed to address the problem of the intent detection problem in the field of natural language processing. In the proposed method, firstly, sense-based vectors are generated from WordNet data using the generalized SemSpace approach, words representing the context of each dataset are clustered in this vector space, and then the disambiguation process is carried out by selecting the closest sense candidate to the context cluster of the words whose sense was ambiguous. Finally, the sense-based contextualized SemSpace vectors we generated are trained with the Bidirectional Long Short-Term Memory (BLSTM) model. To measure the success of the resulting model, tests are conducted on six well-known intent detection benchmark datasets (ATIS, Snips, Facebook, AskUbuntu, WebApp, and Chatbot). According to the comparison results, the recommended method generates context-based word vectors similarly to large language models such as BERT, ELMo, and GPT which are available in the literature, beacuse it uses both sense-based vectors and a context-based word disambiguation method. It is also predicted that this approach can be used successfully in many problems in the field of natural language processing.

Keywords: Intent Detection, Synset Vectors, BLSTM, WordNet, Generalized SemSpace.

ÇUKUROVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

DOKTORA TEZİ

Generalized SemSpace ve BLSTM kullanarak Bağlamsallaştırılmış Niyet Tespiti

Elif Gülfidan TOSUN

Danışman: Prof. Dr. Umut ORHAN

Bilgisayar Mühendisliği Anabilim Dalı

ÖZ

Bu tezde doğal dil işleme alanındaki niyet tespiti problemini çözmek için Contextualized Deep SemSpace adı verilen yeni bir yöntem geliştirilmiştir. Önerilen yöntemde ilk olarak generalized SemSpace yaklaşımıyla WordNet verilerinden anlama dayalı vektörler üretilmiş, kullanılan her veri setinin bağlamını temsil eden sözcükler bu vektör uzayında kümelenmiş ve sonrasında anlamı belirsiz olan kelimelerin bağlam kümesine en yakın anlam adayı seçilerek belirsizlik giderme işlemi yapılmıştır. Son olarak da ürettiğimiz anlama dayalı bağlamsallaştırılmış generalized SemSpace vektörleri BLSTM modeliyle eğitilmiştir. Elde edilen modelin başarısını ölçmek için iyi bilinen altı niyet tespit kıyaslama veri seti (ATIS, Snips, Facebook, AskUbuntu, WebApp ve Chatbot) üzerinde testler yapılmıştır. Karşılaştırma sonuçlarına göre önerilen yöntem hem anlama dayalı vektörler kullandığından hem de bağlama dayalı bir kelime belirsizliği giderme yöntemi içerdiğinden literatürde mevcut olan BERT, ELMo ve GPT gibi büyük dil modellerine benzer şekilde bağlama dayalı kelime vektörleri üretmektedir. Aynı zamanda bu yaklaşımın doğal dil işleme alanındaki birçok problemde başarıyla kullanılabileceği de öngörülmektedir.

Anahtar Kelimeler: Niyet Tespiti, Synset Vektörleri, BLSTM, WordNet, Generalized SemSpace.

GENİŞLETİLMİŞ ÖZET

Dil, insanların aralarında anlaşmasını sağlayan ve kendi kuralları içerisinde gelişen bir yapıdır. Günlük yaşantımızda karşımıza çıkan yazılar ve sesler birer doğal dil örneğidir. Doğal dilde zamanla bazı kelimelerin yok olduğu, bazen de yeni kelimelerin türediği görülmektedir. Bu nedenle insan için bile karmaşık yapıya sahip olan doğal dil işleme sürecinin bilgisayar ortamında işlenmesi oldukça zor bir konudur.

Doğal dil işleme, insanların doğal dilde olan ifadelerini işleyebilmek, analiz ederek kullanışlı sistemler geliştirmek üzerine odaklanır. Literatürdeki çalışmalara bakarak dillerin oluşumu ve doğal dil işleme süreçlerine oldukça yoğun bir ilginin olduğu görünür. Yapay zekâ ve dil biliminin kesişimi olarak ortaya çıkan bu konu özellikle müşteri destek hizmetleri, akıllı ev uygulamaları ve eğitim gibi alanlarda çeşitli uygulamalarla karşımıza çıkmaktadır.

Günümüzde ChatGPT gibi insansı sohbet edebilen diyalog sistemleri doğal dil işleme üzerine ilgiyi artırmıştır. Görev odaklı ve sohbet robotları olmak üzere iki ana başlık altında incelenen diyalog sistemleri özellikle birçok işletme için müşteri hizmetleri alanında çok önemli bir araç haline gelmiştir. Bu sistemler, müşterinin taleplerine hem daha kısa sürede çözümler bulabilmekte hem de müşteri geri bildirimlerini analiz etmeye ve ortak ilgi alanlarının keşfedilmesi gibi pek çok faydaya olanak sağlamaktadır. Ayrıca yapay zekâ destekli bu sistemler ile şirketlerin, müşteriler tarafından yapılan bildirimlere dayanarak yeni ürün önerileri üretmelerine veya tercih ettikleri ürünleri daha iyi anlayarak stratejilerini geliştirmelerine yardımcı olur.

Tüm bu süreçlerin işleyebilmesi için öncelikle kullanıcı tarafından verilen ifadenin anlaşılması ve analiz edilmesi gereklidir. Kullanıcı tarafından doğal dilde verilen ifade ile esas verilmek istenen amacın ne olduğunun bulunması doğal dil işleme alanında niyet tespiti problemi olarak karşımıza çıkmaktadır. Kullanıcının verdiği ifadeye karşılık doğru çıktıların verilebilmesi için kullanıcının niyetinin belirlenmesi oldukça önemli aşamalardan biridir. Genel olarak niyet sınıflandırması olarak da tanımlanan niyet tespiti problemi, kulanıcı ifadesini belirli bir niyetle eşleştirmek için makine öğrenimi yöntemlerini kullanmaktadır. Fakat sistemin öncelikli olarak metin verisini vektörel formatta nasıl temsil edeceği belirlenmelidir.

Literatürde kelimelerin vektörel gösterimleriyle ilgili çalışmalar incelendiğinde çeşitli kelime gömme yöntemleri kullanıldığı görülmektedir. Kelime gömme yöntemleri benzer anlamlara sahip kelimelerin birbirine yakın temsillere sahip olduğu düşüncesine bağlı olarak önceden eğitimi yapılan yöntemlerdir. Vektör veya kelime gömmesi olarak ifade edilen anlamsal uzay kavramı, içerdiği kelimelerin çeşitliliği kadar kullandığı vektör boyutu açısından da önemlidir. Kronolojik sırada one hot encoding ile başlayan kelime vektör gösterimi yöntemleri, çok yüksek boyutlu oluşu, seyrek gösterim dezavantajı ve kelimeler arasındaki benzerliği kodlayamamasından dolayı yerini zamanla bu problemleri dikkate alan yöntemlere bırakmıştır. GloVe, Word2Vec, FastText gibi yöntemler büyük derlemlerde bile kelimelerin 300 gibi makul boyutlu vektörlerle temsil

edilebileceğini göstermiş, ayrıca anlamsal yakınlığa katkısı konusunda da yeni bir bakış açısı kazandırmıştır. Bu yeni bakış açısıyla yapılan çalışmalarda bağlama dayalı yeni yöntemler geliştirilmiş, GloVe ve Word2Vec gibi yöntemlerden farklı olarak kelimeyi içeren bağlama bağlı olarak vektörel temsilleri bulan BERT ve ELMo gibi modeller ortaya çıkmıştır.

Bu tez çalışmasında da BERT ve ELMo yöntemlerindeki bağlamsal anlam vektörlerinden esinlenilerek Orhan ve Tülü'nün (2021) SemSpace yönteminde bazı geliştirmeler yapılmış ve yeni bir yaklaşım tanıtılmıştır. Generalized SemSpace, WordNet'teki her bir synset için anlamsal bir vektör üreten bir yaklaşımdır. Ancak kelimeler cümledeki kullanım bağlamına göre farklı anlamları temsil ettiğinden bazı kelimeler için birden fazla synset adayı bulunmaktadır. İlgili bağlamda hangi anlamının kast edildiğini tespit etmek için, kullanılan veri setinin bağlamının dikkate alındığı bir kelime anlam belirsizlik giderme yöntemi tanıtılmıştır. Kelimelerin bağlam içerisindeki ifade ettiği esas anlamının bulunması, niyet tespiti probleminin başarımını büyük ölçüde etkileyen faktörlerden biridir. Çalışmada da yapılan testler sonucu önerilen kelime belirsizlik giderme yaklaşımının sistem başarısı üzerinde önemli bir etkisi olduğu görülmüştür. Bu amaçla, öncelikle niyet tespiti verisetleri incelenmiş ve sezgisel olarak aynı konu çerçevesinde bir bağlam kümesi olduğu tespit edilmiştir. Ele alınan veri setindeki anlam belirsizliği giderme işlemi, aynı veri setini temsil eden bir bağlam kümesi oluşturularak giderilmiştir. Generalized SemSpace anlam uzayında tespit edilen bağlam kümesi ile anlam belirsizliğine sahip kelime vektörlerinin benzerliği için Öklid uzaklığına dayanan bir yaklaşım benimsenmiştir. Anlamı belirsiz kelimenin en uygun anlam vektörünü bulmak için, seçilen bağlam kümesindeki her bir kelimeye olan uzaklıklar hesaplanmış, böylece bağlam kümesine en yakın aday seçilerek bağlama dayalı belirsizlik giderme işlemi gerçekleştirilmiştir. Veri setlerindeki her cümle için belirsizliği giderilmiş anlamsal vektörlerin üretilmesi sonrası sistemin eğitilmesi aşamasında literatürde oldukça popüler olan BLSTM kullanılmıştır. Kullanılan BLSTM mimarisi kelime gömme, sıralı kelime örüntülerini öğrenme, özellik çıkarma ve sınıflandırma görevlerini yerine getirebilen farklı katmanlardan oluşmaktadır. Kodlama aşamasında sinir ağı için Pyhton içerisinde yer alan Keras kütüphanesi kullanılmıştır. BLSTM katmanlarından embedded boyutu 60 olarak belirlenmiş ve input olan synsetler için 300 boyutlu kelime gömme vektörleri kullanılmıştır. Modelin ara katmanları için ReLU aktivasyonu ve çıktı katmanı için softmax aktivasyonu kullanılmıştır. Verisetindeki lematizasyon, gereksiz sözcüklerin atılması gibi NLP ön işleme adımları için yine Python içerisindeki NLTK paketinden yararlanılmıştır. Son olarak, tüm verisetlerindeki başarıyı belirlemek için 5-fold cross doğrulaması kulanılmıştır.

Çalışmada ayrıca niyet tespiti probleminde başarıyı etkileyen önemli faktörlerden birinin de sözlük dışı (OOV) kelimeler olduğu görülmüştür. Bu amaçla OOV kelimelere, generalized SemSpace vektörlerinin bulunduğu uzayda yeni anlamsal vektörler tanımlanabileceği üzerine odaklanılmıştır. Generalized SemSpace vektörlerinin kelime dağarcığı kullanılan WordNet tanımlarına bağlı olmasından dolayı özellikle kısa cümleli verisetlerinde OOV yüzünden başarının dramatik şekilde düştüğü görülmüştür. Bu dezavantaja rağmen önerilen yaklaşım, verisetlerinin

çoğunda literatürdeki başarıyı yakalamış ve bazılarında da en iyi sonuçları elde etmiştir. OOV probleminin de aşılması sonucu daha da iyi çıktılar elde edileceği aşikardır. Bu açılardan önerilen çalışmanın ileride yapılacak birçok araştırmaya ışık tutacağı düşünülmektedir.

Sonuç olarak bu çalışma kapsamında, kullanılan verisetlerindeki (ATIS, Snips, WepApp, AskUbuntu ve ChatBot) kelimeler için WordNet ağındaki kelime dağarcığına bağlı olarak düşük boyutlu ve yüksek yoğunluklu anlamsal vektörler üreten, BLSTM ağını bu vektörler ile eğitip niyet tespiti probleminde tutarlı sonuçlar elde eden bir model tanıtılmıştır. Başarılı sonuçlar ve literatür karşılaştırmaları, önerilen yaklaşımın çeşitli NLP konularına yeni bir perspektif getirebileceğine dair yeni bir beklenti oluşturmaktadır.

ACKNOWLEDGEMENTS

I would like to present my sincere gratitude to my supervisor Prof. Dr. Umut ORHAN for his guidance, support, motivation and his valuable time for this research work.

I would like to thank to valuable academicians Prof. Dr. Selma Ayşe ÖZEL, Assoc. Prof. Dr. Ali İNAN, Asst. Prof. Dr. Çağatay Neftali Tülü, Asst. Prof. Dr. Erhan TURAN and Asst. Prof. Dr. Mehmet SARIGÜL for their help and support throughout the study.

I would like to thank to The Scientific and Technological Research Council of Turkey (TÜBİTAK) for the BİDEB 2211-A Domestic PhD scholarship numbered 1649B031904212 provided throughout my doctoral education. I am also thankful to Res. Asst. Ferhat ALBAYRAK for his help.

Finally, I would like to thank my husband who has supported me throughout the entire thesis period, both by his love and patience and my family who have supported me throughout my education life.

LIST OF TABLES

LIST OF FIGURES

SYMBOLS AND ABBREVIATIONS

BLSTM : Bidirectional Long Short-Term Memory
CBOW : Continuous Bag of Words
CNN : Convolutional Neural Networks
CRF : Conditional Random Field
DM : Dialogue Manager
GRU : Gated Recurrent Unit
LSTM : Long Short Term Memory
ML : Machine Learning
NLG : Natural Language Generation
NLP : Natural Language Processing
NLU : Natural Language Understanding
NN : Neural Networks
OOV : Out of Vocabulary
RNN : Recurrent Neural Network
R-CRF : Recurrent Conditional Random Field
SVM : Support Vector Machine
WSD : Word Sense Disambigation

1. INTRODUCTION

Natural language consists of spoken and written expressions that spontaneously evolve and come together within a set of rules over a specific period, enabling mutual communication among people living in the same geography. Natural Language Processing (NLP), which first emerged in the 1950s as an intersection of artificial intelligence and linguistics (Nadkarni et al., 2011), is a research field that explores how computers can effectively use natural language texts to produce useful software systems. NLP researchers aim to develop suitable tools and techniques by studying the processes of human understanding and text generation so that computer systems to perform certain tasks based on natural languages (Chowdhury, 2020).

In the early studies of in the field of NLP, scientists tried to model the dictionaries and syntactic rules of natural languages in digital environments. Although serious progress was made in the field, this effort became ineffective beyond a certain point due to the variability and uncertainty of languages. In recent years, factors such as significant increases in computational power, the development of successful machine learning (ML) methods, and the availability of large-scale linguistic data from the internet have provided new perspectives to natural language modeling studies. As a result, NLP has become an important and popular research area.

NLP studies cover a wide range of topics in-depth, from straightforward applications like text classification and sentiment analysis to challenging issues like translation and dialogue systems (Hirschberg and Manning, 2015; Lai et al., 2015; Wang et al.,, 2018; Zhang et al., 2015; Lilleberg et al.,, 2012; Yao et al.,, 2019; Zhang et al.,, 2018; Serban et al.,, 2015, Kawahara, 2019; Cuayáhuitl, 2017). All these popular NLP problems can actually be examined in two main research areas: Natural Language Understanding (NLU) and Natural Language Generation (NLG). On the basis of all NLP tasks interacting with humans, it includes such an important procedure as NLU. The main task of NLU is to perform the process of understanding natural language texts in order to make interpretations from them. It includes a range of tasks including syntactic and semantic analysis, context-based disambiguation, entity recognition, sentiment analysis, and more. Contrarily, the objective of NLG is to produce content in natural languages that humans can understand using structured data, text, pictures, audio, and video (McDonald, 2010).

Dialogue systems, which started with simple text-based systems and today turned into humanoid conversational systems (Hirschberg and Manning, 2015), are systems that define what the other party has said, transform it into a form that the system can understand, and perform the desired action in response to this utterance, conveying it to the other party. In the dialogue system studies which began with the dream of a robot that has the knowledge and skills to chat with humans, there are generally three stages: "NLU", which tries to determine what the given expression means, "DM" (Dialogue Management), which decides the response to be given depending on the conversation

content, and "NLG", which produces a meaningful expression for the decided response. Figure 1.1 (Joukinen and McTear, 2022) shows the general structure of dialogue systems.

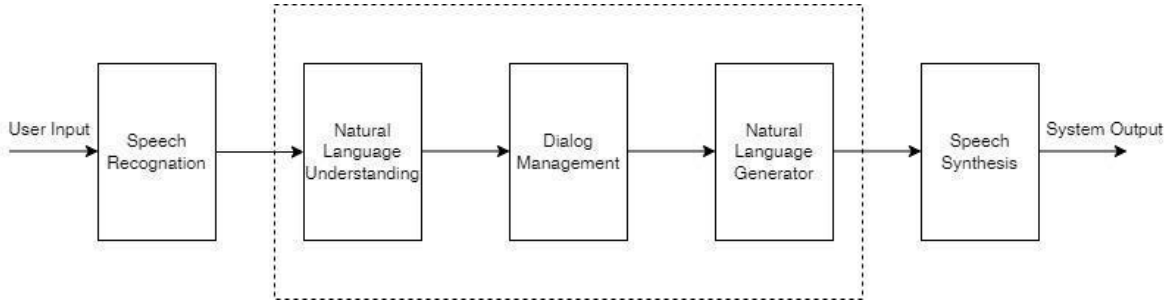

Figure 1.1. General Architecture of Dialogue Systems

NLU aims to extract sense, intent and context from text or speech inputs. NLU systems seek to close the communication gap between humans and machines by allowing computers to interpret and reply to human language in a significant way. When a user utterance is given, the NLU component maps the expression to a structured semantic representation. A popular schema for semantic representation is a dialogue action consisting of intent and slot values. The intent type is a high-level classification that indicates the function of an utterance such as a query or an informative statement. Slot value pairs are also task-oriented semantic elements specified in the utterance. Figure 1.2 shows an example of possible semantic representation for any given utterance.

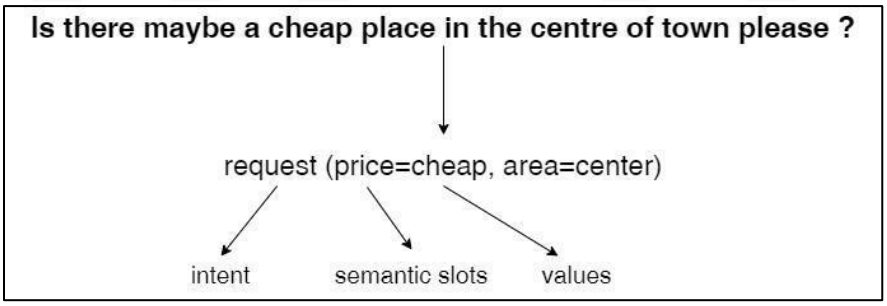

Figure 1.2. Structured semantic representation

The dialogue manager is one of the most fundamental components of the system, managing all aspects of the dialogue. It is considered as the execution controller of the dialogue system and keeps tack of the current state of the dialogue and makes decisions about the system's behavior. NLG, on the other hand, is responsible for generating the output to be given to the user in natural language.

Recently, researchers have seriously focused on two of the most important components of dialogue systems, NLU and DM, and have introduced new methods to improve these components (Ni et al., 2023). A significant part of these method studies has been done on "intent detection" which enables the understanding of the target action in the dialogue (Firdaus et al., 2023). Understanding the main intent behind the user's utterance is highly significant for obtaining useful information from

the expression and providing an appropriate response to the user. Thanks to intent detection, it will be possible to automate many processes and thus provide the most appropriate response for the user.

Intent Detection

Intent detection is one of the problems solved by NLP, which is a subfield of artificial intelligence. The primary purpose is the automatic determination of the underlying intent represented in natural language in a text or speech input provided by the user, in task-oriented dialogue systems. It is widely used in applications such as chatbots, virtual assistants, customer support systems and more, where understanding user intent is crucial to providing relevant and accurate responses.

For example, in terms of customer support systems, intent detection provides various benefits to companies. To briefly talk about these, it helps improve the user experience and understands the intent behind a query or request. In this way, a concrete and relevant response is provided in accordance with the request. This improves customer experience and increases satisfaction. By analyzing customer intent and conversation data, it can assist companies in making informed decisions about their products, services, marketing strategies, and more. It enables a company to increase efficiency in customer service and support processes by responding to a larger number of issues and requests more quickly and accurately.

The intent detection process typically involves a combination of various methods including machine learning and deep learning, to classify input texts into predefined categories or intents. These intents represent the various possible actions or objectives that a user can have when interacting with a system. Intent detection can be defined as a classification problem on task-oriented dialogue systems that can be formulated as shown in Equation 1.1.

$$y_i = i\varepsilon N \; P(y_i|x) \tag{1-1}$$

It seeks to identify the intent of a specific user utterance designated as x, composed of words from the sequence x = $(x_1, x_2, \dots, x_t)$ and categorize it into one of the specified N intent classes as y_i.

When the literature in the field of intent detection is examined, it is observed that studies generally rely on NLP and ML techniques (Ni et al., 2023). In NLP, some preprocessing is also required on the text before feeding the data into any artificial intelligence or statistical method. These preprocessing steps can be listed as cleaning punctuation and meaningless characters, removing stopwords, correcting spelling errors, tokenization, stemming, and lemmatization. These preprocesses help NLP models produce more accurate results.

1.1. The Aims of This Thesis

There are many studies in the literature related to intent detection (Mesnil et al., 2014, et, Zhang et al., 2019, Ren and Xue, 2020). Especially in recent years, it has been observed that studies have focused on context-based response generation methods such as BERT (Devlin et al., 2018), ELMo (Sarzynska-Wawer et al., 2021), and GPT (Radfor et al., 2019) and have achieved successful results. Starting from this, in this study, it is aimed to produce both sense-based word vectors and to present a method that takes context into account. In the generation of sense-based vectors, inspiration is drawn from Orhan and Tulu's (2021) SemSpace method, and at the stage of the inclusion of context in the method, a disambiguation approach based on the detection of the context cluster defined in the generalized SemSpace semantic space is proposed. In addition to the obtained results being promising in terms of the intent detection problem, it is thought that it is inevitable that the use of this approach in the slot filling task, which is another important stage of NLU, will contribute to the system accuracy.

1.2. Our Contribution

Studies on intent detection started in the early 2000s, and in these studies, machine learning methodologies were applied to natural language (Kim et al., 2016, Orhan et al., 2023, Firdaus et al., 2023). In this study, firstly, synset vectors were created from WordNet data with a low-cost calculation, the calculated vectors were used for context analysis in the disambiguation processes of the selected datasets for intent detection tests and finally, an approach that performs intent classification using a supervised deep learning model was proposed. The suggested method is able to provide context-based word vectors identical to those produced by the BERT, ELMo, and GPT approaches because it makes use of both sense-based vectors and a context-cluster-based word sense disambiguation (WSD). The method is named as “contextualized deep SemSpace” because it can generate context-based vectors and is inspired by the SemSpace approach proposed by Orhan and Tulu (2021).

The main idea is to train the proposed method by organizing the synsets and relationships in the WordNet 3.1 dataset and determine the sense vectors corresponding to each synset after training. After this process, appropriate sense vectors corresponding to each word in the dataset to be used were found by applying WSD and classified using the Bidirectional Long Short Term Memory (BLSTM) model. All these studies were tested using well-known and commonly used intent detection datasets in the literature. Although the dependence of the vocabulary on WordNet causes the problem of eliminating a significant number of words because their vectors cannot be found, the results have shown that the proposed approach is one of the most successful intent classifiers in the literature. These findings provide strong evidence that contextualized synset vectors based on deep learning can be used successfully to a variety of issues.

1.3. Thesis Organization

The remaining sections of the thesis work are organized as follows:

The second chapter reviews the literature and provides a summary of significant studies that have contributed to the field of intention detection. Solution methods found in the literature, traditional methods and methods using machine learning is scanned and these studies is included.

In the third chapter, the materials and method are explained. All data sets used to carry out the thesis study are described and all methods used are explained. At the end of the chapter, the proposed “Contextualized Deep SemSpace” intent detection method is explained in detail.

In the fourth section, the results of the proposed method are presented in detail. Measurement results obtained with benchmark data sets are analyzed and compared with successful methods in the literature.

In the last chapter, comments and suggestions regarding the results obtained in this study are included. Following a discussion of the benefits and drawbacks of the suggested approach, the outlook for the future is presented.

2. PRELIMINARY WORK

In many different fields, the intent detection task is a crucial part of dialogue systems. It's critical to understand the intent in user expressions for automating operations and obtaining important context-related data. In the literature, statistical (Dowding et al., 1994; Etzioni et al., 2004) and machine learning-based (Xiu and Sarikaya, 2013; Lai et al., 2015; Ayanouz et al., 2020) approaches are used for intent detection tasks. Although traditional statistical methods are used, these methods are not capable of learning deep semantic knowledge. Studies using deep neural networks have brought new insights into various NLP tasks, containing intent detection and slot filling tasks (Firdaus et al., 2021; Weld et al., 2021). In these studies, machine learning methods are generally used after the words are converted into vectors. Recurrent neural network (RNN) techniques are the most widely utilized machine learning models (Lai et al., 2015; Xu and Sarikaya, 2013).

Etzioni et al. (2004) developed a system called 'KNOWITALL', which aimes to automate information extraction from large-scale corpora. The proposed system is a scalable and open-domain web-based question-answering system that evaluates the information extracted using statistics calculated by processing the Web as a large text corpus. Since the method used in the study is aimed to be easily scalable for new classes and to minimize the manually entered training data set, the bootstrapping method is used to reveal the result by using general inference patterns and automatically generated distinguishing expressions. The Bootstrap method involves resampling to create larger-scale datasets from the existing dataset (Sacchi, 1998). Another distinguishing feature of the system is the use of PMI-IR (Pointwise Mutual Information-Inverse Retrieval) methods to evaluate the probability of inference using ‘web-scale statistics’. PMI is a scale that has been shown to meet the conditions of scalability, incrementality and simplicity and has demonstrated high performance in compulsory selection tests for semantic similarity (Recchia, 2009). Additionally, since the system consumes the output of existing search engines, it also has the advantage of automatically benefiting from improvements made on search engines on the web.

Tur et al. (2005) aimed to reduce the effort of labeling for NLU by using a combination of active and semi-supervised learning methods instead of training using expressions labeled by a large number of people, whose preparation is labor and time consuming. The goal of the active learning algorithm is to select examples that will provide the greatest improvement in performance, thus reducing the amount of human labeling effort. In short, active learning determines which data needs to be labeled, and semi-supervised learning determines what to do with the remaining unlabeled data. The first semi-supervised learning method increases the training data using machine-labeled classes for unlabeled expressions. The second method, on the other hand, again trains an initial model using some human-labeled data and then incrementally enhances a classification model trained using human-labeled expressions with machine-labeled ones. In this study, a Boosting-style classification algorithm is used for classification. Boosting aims to combine weak base classifiers into a strong

classifier. This method's iteration involves training a weak classifier on a weighted training sets, and eventually the weak classifiers are integareted into a single composite classifier. The goal of semi-supervised learning is to take advantage of unlabeled expressions to improve the performance of the classifier. In the results obtained, it is observed that the same classification accuracy could be achieved by using less than half of the labeled data.

Cao et al. (2011) developed a question-answering system called 'AskHERMES' that performs semantic analysis of complex clinical questions. Instead of presenting a single true answer or a list of documents in response to clinical questions, the system aims to improve the quality of answers by extracting the most appropriate information for a particular question. Unlike most NLP studies that focus on textual in the form of explanations, this study also considered information in tables and lists. When compared with the Google search engine and the upToDate database system, it is observed that the developed system is comparable to systems such as Google and upToDate in terms of ease of use, answer quality, time spent and general performance criteria.

Tur et al. (2011), in their previous studies on NLU error analysis revealed that the most common causes of errors in slot filling and intent detection tasks are due to unimportant syntactic features. To address this, they proposed a dependency parsing-based sentence simplification approach that extracts a set of keywords from natural language sentences and uses them in addition to all expressions to complete NLU tasks. For example, in dependency parsing created for the "I need to fly from London to Boston," sentence, the top-level token is 'fly'. Its dependents are 'from' and 'to'. The sentence is then simplified as follows: "fly from to." When the results obtained are examined, it is observed that head/dependencies pairs are not contribute significantly to intent detection performance, but using the predicate instead of immediate head is very important in reducing the error rate for slot filling tasks.

In their study, Heck and Tur (2012) proposed an unsupervised training approach for NLU systems that uses semantic knowledge graphs over the web. This approach generates natural language forms of entity-relation-entity sections using a combination of web search retrieval and syntax-based dependency parsing. The semantic representation guides the creation of entity-relationship patterns used to extract natural language surface forms from the web. These surface forms are used to automatically train NLU systems in an unsupervised manner, covering the information represented in the semantic graph. The study used the state-of-the-art "Berkeley Parser" for parsing and the LTH Constituency-to-Dependency Conversion toolkit to create `dependency parses ` from the created `output parse trees`. Using these enriched semantic graphs, it has been concluded that NLU labels can be extracted automatically from the training data, by training the NLU systems with automatic labels in an unsupervised way, it can match the supervised training performance in the intent detection problem.

Chen et al. (2013) proposed an unsupervised approach that utilizes the "frame semantics" theory to automatically create and fill semantic slots on unlabeled data. In this approach, frame

semantic parses are generated using a statistical probabilistic semantic parser (SEMAFOR) trained by FrameNet. Since SEMAFOR, the parser used in the study, is trained on FrameNet which has a more general frame semantic content, not all frames in the parsing results can be directly used as real slots in open-domain dialogue systems. FrameNet style frame semantic parsings need to be mapped to semantic fields in the target domain. Chen et al. (2013) formulated this problem as a re-ranking problem and suggested the use of an unsupervised spectral clustering-based slot ranking model to solve it. As a result of the study, it has been observed that the automatically extracted semantic slots and the reference slots created by the field experts are compatible.

Xu and Sarıkaya (2013) presented a neural network based Conditional Random Field (CRF) architecture for intent detection and slot filling. This joint model, which is based on convolutional neural networks (CNN), is a neural network (NN) adaptation of the triangular CRF model (TriCRF), in which the dependencies between the intent label and slot sequence are taken advantage of. Although there are studies in the literature on a similar approach, the approaches presented in previous studies are limited to using linear-chain CRF models, and the features are not derived from word sequences of variable length. In the proposed model, features are extracted from CNN layers, and the slot filling process is performed by the CRF model. The aim of this study is to take advantage of the powerful feature learning mechanism provided by NNs and apply it to joint intent and slot modeling for NLU. In summary, the presented architecture extracts features in CNN layers and the resulting feature vectors are shared by the two tasks. In this way, both the universal nature of the labeling component and the system can perform multiple labeling and classification tasks at the same time. This study demonstrates the advantage of joint modeling within the proposed NN architecture compared to the standard TriCRF model, and it has been observed that the slot model component produces results that outperform CRF significantly.

In the reviewed literature, it is concluded that simple RNNs and CNNs performed better than CRFs. Building upon this, Yao et al. (2014) investigated the performance of advanced Long Short Term Memory (LSTM) neural networks which are more advanced than simple RNN, on labelling words on the ATIS dataset. They extended the basic LSTM architecture to a structure that explicitly models the dependencies between semantic labels by using outputs at time t-1 and time t steps and performing a regression operation on predictions. At the same time, in order to evaluate which gates in LSTM models are important for NLU tasks, they simplified LSTM models by only keeping certain gates and compared the performance of these simplified models with the classic LSTM. The simplest modification to evaluate the importance of gates in the NLU is to ignore all gating functions. However, in this case, memory cells accumulate a history of input without discarding past memories. The input and outputs are not modulated to memory cells. Therefore, simplification can be achieved by fixing one gate to 1.0 and learning the other two gates. When the F1 scores in the study were evaluated, it is concluded that if two gates are to be added to the simplified LSTM, one of these gates should be the forget gate. As a result of the study conducted on the ATIS dataset, it was observed

that LSTM performed better than simple RNNs in finding and utilizing long-range dependencies in the data.

Kim et al. (2016) used the bidirectional LSTM (BLSTM) in their intent detection study. In contrast to earlier works, on input vectors, they aimed to enrich word vectors using semantic vocabularies like WordNet. In addition to learning embeddings within the neural network, they created a BLSTM architecture by enriching the embeddings outside the neural network. They also trained their models on datasets obtained from Microsoft Cortana and ATIS.

In the NLU stage, which is one of the most important components of dialogue systems, the semantic parsing of input expressions typically consists of three tasks. These are domain detection, intent recognition and slot filling. Semantic slot filling is one of the challenging problems in this stage. For this purpose, Mesnil et al. (2015) compared standard RNN architectures such as Elman and Jordan, as well as hybrid, bidirectional and CRF extensions. One of the extensions they explored is the R-CRF (Recurrent Conditional Random Field) structure, which uses the CRF's objective function and trains RNN parameters according to this objective function. R-CRF is different from previous work that used feedforward neural networks and CRFs with CNNs. While Xu and Sarıkaya (2013) used CNNs in feature extraction, they used RNNs. The results of their study confirmed that R-CRF improves on a simple RNN. Generally, the findings indicated that both Elman-type and Jordan-type networks outperformed CRF significantly. Additionally, it is also shown that a bidirectional Jordan-type network that takes into account both past and future dependencies between slots works best.

Sarıkaya et al. (2016) discussed Microsoft's digital assistant, one of the digital assistant systems such as Siri, Google Now and Alexa, which have become the primary focus area for large technology companies. And they described the architecture of this system's NLU and dialog management components. Implementing an open-domain dialog system is difficult because the users are not limited in what they can say. In some studies, open-domain dialog systems have been implemented for each field, with language understanding and dialogue managed by independent field expert systems. In contrast to these similar studies, in the system described by Sarıkaya et al. (2015), in order to develop a open-domain system, language understanding and some dialogue components are centralized and the system's support for new domains was improved. The system covers a wide spectrum such as searching the internet, answering questions, setting alarms and timers, playing music. The designed architecture adopted a layered approach. The base layer provides a web search service that serves queries that cannot be processed by more domain-specific providers. The next layer is question-answering services that cover various domains and provide answers without requiring users to navigate directly to relevant websites. To provide a more complex query, this layer can benefit from obtaining contextual information from previous turns. In the NLU stage, they presented the query in a semantic framework consisting of <DOMAIN, INTENT, SLOTS> tuples

and used machine learning classifiers such as Support Vector Machines (SVM) for domain classification. CRFs and newer deep learning methods have been used for slot labeling.

The slot filling task in NLU involves extracting relevant semantic components from natural language expressions. For this purpose, Celikyilmaz et al. (2016) proposed an a semi-supervised CNN2CRF architecture approach that works on unlabeled data, incorporating a CNN structure with CRF for model training. In this approach, CNN is used as the bottom layer to learn feature representations from labeled and unlabeled sequences, and CRF is the top layer. Two CRF structures are used in the top layer. The first CRF model updates its weights only with labeled training data, the second CRF model updates its weights with unlabeled expressions. Briefly, with this method, a joint pre-training and NN classification that learns the network from both labeled and unlabeled data is presented. The proposed pre-training method with the CRF2CNN architecture results in comparable performance according to the strongest semi-supervised baseline in slot filling tasks has been achieved, even without extensive manually labeled training data.

The intention of a sentence and its semantic slots are relational. From these two tasks, information from one task can be used to introduce the other task, and a joint prediction can be made. Based on this idea, Zhang and Wang (2016) developed a joint model for both tasks in their study. They used Gated Recurrent Unit (GRU) to capture the representation of each time step in a sequence of sentences. The representations obtained here are used for predicting slot labels, and at the same time, a universal representation of the sequence for the intent detection task is learned using a max-pooling layer. Consequently, the representations learned by GRU are shared between the two tasks. The universal representation is obtained by maximizing the shared representations to predict intent labels. A comparison is made for the proposed model based on SVM, CRF, RNN, R-CRF, Boosting and it is concluded that the proposed approach performs better than these approaches.

Korpusik and Glass (2017) proposed a new approach to food diaries that utilize speech and NLU technology to enable efficient evaluation of energy and nutrient consumption for food recording used in the prevention and treatment of obesity. In the proposed system, after the user identifies their food, the NLU component should not only identify the foods and their features but also determine which foods are associated with which features. For example, in the sentence "I had a bowl of cereal and two cups of milk," after labeling 'bowl' and 'two cups' as 'quantity' and 'cereal' and 'milk' as 'food,' the system should be established that the expression "two cups" has a relationship with "milk" as the food it describes. For semantic tagging, they initially applied a standard CRF basic model to which more complex feature sets containing word vectors are added. Then they examined the solution of the semantic labeling problem using a CNN. In the process of creating word vectors, they used the Continuous Bag of Words (CBOW) approach, which predicts the current word based on the context. In their comparisons, it is observed that CNN outperformed CRF and provided better performance for semantic tagging.

Firdaus et al. (2021) proposed a deep learning-based multitasking model that can perform both intent detection and slot filling tasks together. In their multitasking model, they used a deep bidirectional RNN architecture with LSTM and GRU as the frameworks.

Similar to the majority of studies in the literature, Zhao et al. (2022) have presented a hybrid model for intent detection and slot filling tasks. They developed a program structure that is less linear and more effective at parallel execution than the learning models in the literature. The suggested approach is a recurrent, convolutional neural network model combined with a multi-head attention mechanism.

Sun et al. (2022) have proposed a model that combines two important tasks of the NLU system, which are slot filling and intent determination. Many models proposed in the literature are joint models that focus on only surface sharing parameters instead of bidirectional interactions. However, in this study, a dual interaction model based on the gate mechanism is suggested by them. They used a Self-Attention Extended Deep Convolutional Neural Networks (DCNN) block to better capture the meaning of expressions. They adopted the gate mechanism to obtain interaction information between intent and domain that can fully utilize the semantic relevance level between slot filling and intent detection. They evaluated the proposed model on the ATIS and SNIPS datasets, which are widely used in intent detection and slot filling studies in the literature.

In order to handle both issues simultaneously, Hui et al. (2021) suggested a Continuous Learning Interrelated Model (CLIM) model that takes into consideration the information with various characteristics needed for both tasks. As a result of their work on Atis and Snips datasets, the model reached the latest performance.

A hybrid model for intent recognition and slot filling that uses a multi-stage hierarchical approach using BERT and a bidirectional NLU mechanism is suggested by Han et al. (2022). The bidirectional mechanism used to facilitate information exchange between intent recognition and slot filling consists of Intent2slot and slot2intent models. The Intent2slot model uses the semantic information of the entire expression when labeling each word. On the other hand, the probability distributions of expression labels, are taken by the Slot2intent model as supplementary data for intent detection. The model evaluation is conducted on ATIS and Snips, which are well known in the literature.

Ma et al. (2022) stated that although slot types are semantically related to intent, slot positions of the same intention can differ greatly in different utterances due to the diversity of spoken language. Therefore, they argued that the traditional single-stage slot filling task may provide irrelevant information for slot position prediction in slot-intent interactions in joint models. As a solution, they proposed a new two-stage selective fusion framework for joint intent detection and slot filling. In contrast to the traditional single-stage framework, the proposed framework divides slot filling into two stages: slot proposal and slot classification. A slot proposal network, consisting of

BERT and BLSTM – CRF, predicts slot positions. The proposed model is tested on well-known datasets in the literature, including ATIS, SNIPS, MixATIS, MixSNIPS, 24, and DSTC4.

Chen et al. (2022) addressed the issue that traditional attention is insufficient because it only captures the first-order feature interaction between two tasks. In this regard, they proposed a unified BiLinear attention block to capture second-order interactions between intent and slot features. This model utilizes bi-linear pooling to concurrently explore both contextual and channel-wise bi-linear attention distributions. Higher order interactions are formed by combining multiple such blocks and exploiting Exponential Linear Unit activations. Additionally, they introduced a High-Level Attention Network (HAN) to jointly model these interactions.

Wei et al. (2022) have proposed a Wheel Graph Attention Network (WheelGAT) model that can directly model interrelated connections for intent detection and slot filling. Similar to recent studies, they have presented a joint model that deals with intent detection and slot filling tasks together. According to the proposed model, a graph structure is created for utterances, consisting of intent nodes, slot nodes, and directed edges. They have stated that intent nodes can provide semantic information at the utterance level for slot filling, and slot nodes can provide local keyword information for intent detection.

Zhang et al. (2022) have focused on the problem that many methods available in the literature cannot fully utilize the limited examples when classifying unseen classes, mainly because they rely on the generic knowledge acquired in base classes to define new classes. In this regard, they have proposed a Contrastive Learning based Task-Adaption model (CTA) for multistep intent detection. Firstly, in the task-adaption module, Using a self-attention layer that aims to establish interactions between examples from different categories, they based the new representations on task-specific rather than base classes. They have used comparison-based loss functions and label name semantics to reduce the similarity between sample representations in different categories while increasing those in the same categories, respectively. They have used the OOS, SNIPS, and ATIS datasets in their training experiments.

Lu et al. (2023) have proposed a method to integrate the semantic information of intent and slot labels into a joint model. Firstly, they converted intent, slot labels, and "B, I, O" labels into natural language to take full advantage of the semantic knowledge of intent and slot. Subsequently, they used the pretraining model BERT to encode natural language and interact with the encoded representation of the sentence. Finally, they performed their training on three datasets, namely ATIS, SNIPS, and Facebook.

Most of the existing joint models that handle both intent detection and slot filling tasks together build two different decoders on top of a shared weight encoder or utilize intent information to detect slots. In some cases, information is transferred indirectly between two tasks. However, Tu and et al. (2023) have proposed a bidirectional joint model in their study that explicitly incorporates intent information into slot filling and slot information into intent detection for NLU. Firstly, a

temporary intent is predicted, which is fed to a biaffine classifier to recognize slots, and then the slot features are used together with the expression representation to predict the final intent. They have also presented a loss function that takes into account three different losses: temporary intent detection, final intent detection and slot filling. They performed their tests on the ATIS, Snips, and PhoATIS datasets.

Firdaus et al. (2023) have addressed the multilingualism problem in NLU systems and proposed a multilingual multitasking approach to combine intent detection and slot filling for three different languages. Since intent detection and slot filling tasks are both highly interrelated, they proposed a multitasking strategy to address these two tasks simultaneously. They used a converter as a shared sentence encoder for English, Hindi and Bengali languages. Their study is conducted on the Atis, Snips, Frames and Trains datasets.

When the literature related to NLU is examined, the use of neural networks with embedding layers, where word vectors are integrated, is noteworthy. Word vectors utilized in these techniques are frequently created beforehand using word embedding or sense embedding techniques. Word embedding is a technique that models the cooccurrence of lexical units in a corpus and singularizes them (Mikolov et al., 2014; Pennington et al., 2014; Joulin et al., 2016; Bojanowski and et al., 2017). However, word embedding techniques, only assess words lexically, therefore they combine all of a word's meanings into a single vector. In sense embedding approaches that aim to address the drawbacks of these methods by combining semantic networks with corpus data, various vectors are constructed for each meaning of the words (Pantel and Lin, 2002; Chen et al., 2014; Li and Jurafsky, 2015; Jauhar et al., 2015; Rothe and Schütze, 2015; Camacho-Collados et al., 2016; Song, 2016; Camacho-Collados and Pilehvar, 2018; Sinoara et al., 2019). State-of-art embedding techniques like BERT, Elmo, and GPT produce better outcomes than other embedding techniques, because they generate context-dependent word vectors. However, to obtain a "pre-trained model," these methods must train on a very big corpus, which needs expensive hardware and a long processing period. If some domain-specific words are not present in this "pretrained model", researchers need to repeat this pretraining process to identify such words into the model. Nevertheless, this is a costly training procedure that not everyone can afford to perform.

In contrast to the studies in the literature, Orhan and Tulu (2021) proposed the SemSpace method, in which word sense vectors can be detected in Euclidean space using only WordNet (without using corpus), with a special approach that assigns weights to sense relations. It is stated that in some datasets where human-assessed word similarities are determined, the best results are achieved using the sense vectors determined with the SemSpace method. In this study, we focused on intent detection, which is one of the most important steps in the NLU field, and a contextualized deep SemSpace approach was proposed based on Orhan and Tulu's (2021) SemSpace method. Firstly, an Euclidean semantic space based on the generalized SemSpace method is defined, and synset cluster vectors are generated from WordNet data with a low-cost computation. According to

the dataset utilized for intent detection, the proposed method contextualizes the vectors. Finally, the dataset's intent classes are used as output after training with a deep learning model like BLSTM utilizing the determined synset vectors as input. To evaluate the effectiveness of the method, as in most studies in the literature, ATIS, Snips, AskUbuntu, Facebook, WebApp, and ChatBot datasets which are well-known datasets in the intent detection problem, are used.

3. MATERIAL AND METHOD

This study introduces a methodology that comprises three distinct phases. Firstly, relations and synsets in WordNet 3.1 data are trained with the generalized SemSpace method, following this training, each synset is assigned a vector in Euclidean space. The utterances in the dialogue datasets that will use for testing are tokenized in the second stage, then the correct synset candidates for these tokens are found. For this, a word sense disambiguation algorithm based on the nearest neighbor to the context cluster is applied. Lastly, a BLSTM model is used to categorize each utterance in the datasets. All of these studies are conducted using intent detection datasets that are frequently used in the literature.

The datasets and methodologies used in the study are thoroughly detailed in this part of the thesis.

3.1. Materials

In this section, firstly, the WordNet dataset used to prepare sense vectors will be mentioned, and then the technical details of ATIS (Airline Travel Information System), Facebook's Multilingual, Snips, WebApp, AskUbuntu and Chatbot datasets, which are widely used in intent detection studies, will be given.

Princeton English WordNet: WordNet (Miller, 1995; Fellbaum, 1998) is a large graph-based dictionary database created based on English concepts. This structure emerged at Princeton University in 1986 and is still being developed. It contains semantic concepts and the relationships between these concepts. Nouns, adjectives, verbs, and adverbs are grouped into sets called synsets, each of which expresses a different concept. Synsets are collections of synonyms, which are words that express the same concept and can be used interchangeably in various contexts.

The most obvious difference between WordNet and a standard dictionary is that WordNet categorizes words into five categories: nouns, verbs, adjectives, adverbs, and function words. While standard dictionaries define concepts by referring to words, WordNet references concepts and presents words within concepts. Another feature is that WordNet connects concepts. All concepts defined in WordNet are connected to each other through 26 different types of relationships, including antonym, hypernym, hyponym and more. Each concept can be linked to other concepts through one or more relationships. Therefore, a word defined in WordNet can be seen in multiple synsets, and with this structure, WordNet has a graph-based design. A graph representation of part of the WordNet network is shown in Figure 3.1 (Miller, 1990).

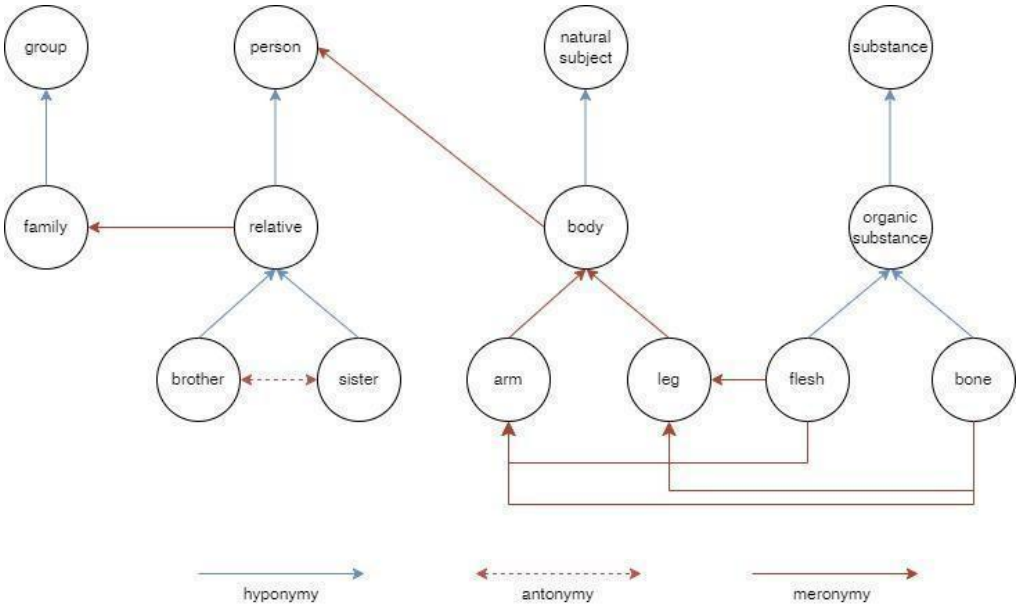

Figure 3.1. Network representation of three semantic relationships

In the graph structure shown in Figure 3.1, the nodes represent synsets and the edges show the relationships between sense pairs. Examples of semantic relationships used by WordNet include synonymy, hyponymy, meronymy, similarity, domain and cause etc. Some relationships are used for word form relationships, while others are used for semantic relationships. For example, in Figure 3.1, the term "leg" has a meronymy relationship with the term "body" because "leg" is a part of "body." Similarly, the term "relative" is a hyponym of the word "brother" because it has a more general sense.

ATIS: An important by-product of the DARPA speech recognition project (Tur et al., 2010) is the ATIS corpus. The ATIS (Hemphill, 1990) is a dataset that is commonly used in NLU researches, contains a large amount of expressions and their associated intentions and can be used in classification problems. The ATIS data set, prepared from voice recordings of flight reservations, consists of 26 distinct intent types and a total of 5,871 utterances of which 4,978 in the training set and 893 in the test set. These utterances cover a wide range of topics related to air travel, such as flight reservations, airport information, ticket prices and more. Each query is explained with two main components:

- Intent: It represents the general intent or goal of the user's query. It shows what action the user wants to perform, such as making a flight reservation, requesting flight schedules or searching for airport information. The intent serves as a high-level classification label for the query.
- Slot: They are specific pieces of information that the system needs to extract from the user's query in order to fulfill the user's request accurately. For example, when a user

wants to make a flight reservation, the slots will include the departure city, arrival city, travel date and other relevant details. Slot filling is an important task in NLP that involves identifying and extracting these important pieces of information from the user's utterance.

Snips: Snips is a dataset collected from a personal voice assistant that contains a labeled form of various questions asked by users in natural language (Coucke et al., 2018). It is used in the fields of NLP and speech recognition, consisting of a total of 7 different intents and 13,784 examples. These questions are marked with labels covering different intents such as weather, reservation requests, music playback requests and more. It contains labeled examples of user queries with their associated intents and slot information.

Faceook's Multilingual: It is a dataset that contains annotated expressions in three different languages: English, Spanish, and Thai (Schuster et al., 2018). The English-language dataset used for this study includes 12 different intents and 38,841 expressions.

AskUbuntu: It consists of 4 different intents (Setup Printer, Make Update, Software Recommendation, Shutdown Computer) and 154 examples extracted from the AskUbuntu platform. There are 10 training 37 test statements for the 'Make Update' intent, 10 training 13 test statements for the 'Setup Printer' intent, 13 training 14 test statements for the 'Shutdown Computer' intent and 17 training 40 test statements for the 'Software Recommendation' intent (Braun and et al., 2007).

WebApp: This dataset has similar characteristics with the AskUbuntu dataset and is constructed in a similar manner. It consists of 7 different intents (Delete Account, Change Password, Download Video, Filter Spam , Export Data, Find Alternative, Sync Accounts) and 83 sample expressions. There are 2 training 6 test statements for 'Change Password' intent, 7 training 10 test statements for 'Delete Account' intent, 1 training 0 test statements for 'Download Video' intent, 2 training 3 test statements for 'Export Data' intent, ' 6 training 14 test statements for the 'Filter Spam' intent, 7 training 16 test statements for the 'Find Alternative' intent, and 3 training 6 test statements for the 'Sync Accounts' intent (Braun and et al., 2007).

AskUbuntu and WebApplication are Stack Exchange site platforms where users may ask and answer questions, also vote on both questions and answers. They are developed by several Question-Answering (QA) groups (Dos Santos and et al., 2015).

Chatbot: The Chatbot corpus is a dataset consisting of a total of 205 questions and 2 different intents, collected by a Telegram chatbot in the production area and answers questions about public

transportation connections (Braun and et al., 2017). Table 3.3 displays the numbers for each intent in the test and training sets.

Atis and Snips datasets are available AskUbuntu, WebApp, and Chatbot datasets are available on GitHub under the Creative Commons.[1] Although the described data sets are presented in two groups as training and testing, these two parts are combined since k-fold validation is used in this study. During the verification phase, grouping was made. Table 3.1 shows a few[1] utterances of the expressions and their intents in the six data sets used.

Table 3.1. Examples of the 6 data sets used

Dataset	**Intent**	**Utterance**
ATIS	flight	all flights from boston to washington
	aircraft	what types of aircraft does delta fly
	airfare	show me economy fares from dallas to baltimore
Snips	GetWeather	please let me know the weather forcast of stanislaus national forest far in nine months
	PlayMusic	listen to westbam alumb allergic on google music
	AddToPlaylist	add slimm cutta calhoun to my this is prince playlist
Face book	weather/find	is monday to be sunny
	alarm/cancel_alarm	tell me my alarms
	reminder/set_reminder	i don't want to forget to eat my lunch
Web App	delete account	how to delete my imgur account
	export data	how can i backup my wordpress com hosted blog
	filter spam	stopping spam emails in gmail by pattern
Ask Ubuntu	make update	How to upgrade from Ubuntu 10:10 to 11:04
	setup printer	How do I install drivers for the Panasonic MB1900CX All-in-One Printer/Scanner
	shutdown	How can one shutdown a PC using the keyboard
Chat bot	FindConnection	what's the cheapest way from neuperlach süd to lehel
	DepartureTime	when does the next bus leaves at garching
	FindConnection	can you find a connection from garching to hauptbahnhof

[1] https://github.com/sebischair/NLU-Evaluation-Corpora

Although each dataset is used separately in the proposed model, for convenience of display, Table 3.1 shows three instances from each of the six data sets. When textual utterances are examined with the human eyes, it is noticed that sentences often include words referred to as named entities. It can also be asserted that these words often play a crucial role in comprehending the intent. The fact that some of these words are not present in the WordNet dataset utilized for the generalized SemSpace training is notable.

3.2. SemSpace Algorithm

The SemSpace algorithm was first developed by Tulu (2019). It is a word embedding method develop to identify word vectors of synsets in WordNet data and discover the optimal weights for sense relationships. SemSpace first discovers their optimum weights by aligning the semantic relationships in WordNet to the values generated by human intelligence, then identifies the word vectors to correspond to the words in the synsets by adjusting the Euclidean distances between them. It focuses on determining word vectors by using prototype-based reinforcement learning methods to calculate similarity between words. It requires two important inputs: a lexical-semantic network like WordNet and a lexical-level similarity dataset created by humans. In this method, WordNet 3.0 data is used for the lexical-semantic network, and RG65, WS353 and MEN300 datasets are used to align the semantic weights. The MEN3000 dataset is commonly used to calculate semantic weights for the following reasons:

- The example list is generated by experts,
- The evaluations of word pairs are made by native English speakers
- The data set contains a sufficient number of word pair examples and represents real-life scenarios.

However, a strength of this work is the use of WordNet, an existing and well-defined lexical-semantic network created by professionals, to establish semantic weights. The comparison of the weights of semantic relationships using a common and significant data set like RG65 and the use of a sizable dataset like MEN3000 for development and testing purposes both offer a new perspective on the calculation of semantic similarity and semantic relationships.

In addition to generating semantic word vectors, another crucial aspect of the SemSpace method that it allows obtaining new word pairs and similarity scores in WordNet. An example of this scenario is shown in Figure 3.2.

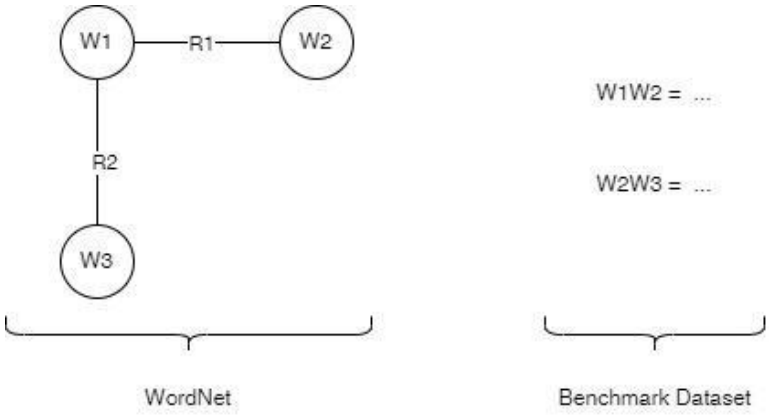

Figure 3.2. Indefinite relationships

Figure 3.2 shows W_1W_2 and W_1W_3 pairs related in WordNet. When examining the benchmark dataset, it becomes clear that some W_2W_3 relationships that are not found in WordNet are present in these datasets.

For the SemSpace method, all words in WordNet are first transformed into independent nodes in a graph structure separate from the synset structure. In the algorithm, each word is considered as a prototype in the hyper space, and the relationships between words that show how similar they are, are described as equations with two unknowns. However, as shown in Figure 3.3, it is clear that there can be an infinite number of solutions with three unknowns and two equations.

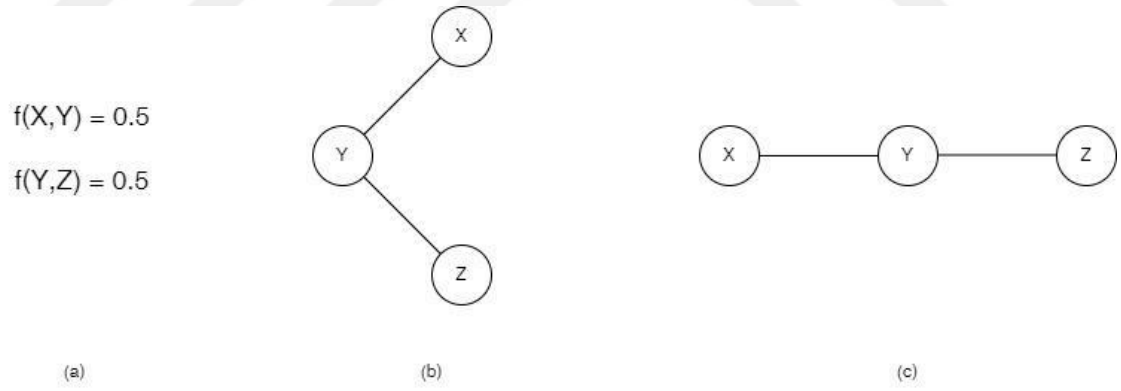

Figure 3.3. Example solutions with two unknowns

An example of the converted graph structure of WordNet is shown in Figure 3.3. To model it as a graph, each variable is indicated as a node and the relationships between any two variables are represented as edges. In Figure 3.3, X, Y, and Z correspond to words in WordNet, and there are relationships X-Y and Y-Z. The functions f(X, Y) and f(Y, Z) represent similarity scores associated with these relationships. Possible solutions for the given two functions are shown in Figure 3.3 (b) and Figure 3.3 (c). Although both of them are possible solutions, they are quite different from each other. Therefore, it seems that even in just two-dimensional space, there are countless solutions. On the other side, infinite solutions can be created by rotating solution models in space as the dimension

of the hyper space rises or the geometric angles between nodes change. In such problems, the system must be transformed into a complete graph that includes every node to use the term "the best single solution", which is not feasible in the majority of real-world problems. Therefore, in the absence of a sufficient number of equations, the best approach is to look for a successful solution according to the intent function with an intuitive approach.

Each word in the SemSpace technique is represented as a prototype point in the prototype space, which is influenced by Learning Vector Quantization (Kohonen, 1995). Each similarity relationship between two words is treated as an equation with two unknowns. The algorithm also acts as a search algorithm with random additions and performs learning by repeating it many times. In the method, it is posited that once the optimal positions for prototypes are found, the system will generate the best correlation and the smallest deviation. For this, the algorithm requires separate training and test sets. For the SemSpace algorithm, datasets such as RG65, WS353, and MEN3000 have been used, but the word pairs obtained from WordNet cannot be used directly in the study because they are associated with directional relationships rather than numerical similarities. To overcome this issue, weights between [0, 1] and relationship types defined in WordNet have been identified. First, a weighting procedure is used to get the best values for these weights.

WordNet Weighting

The first step for the SemSpace method is the preparation of the training dataset from lexical-semantic WordNet used. In WordNet, each word is separated from the synset idea. As shown in Table 3.2, the relationships in WordNet are encoded with single characters to facilitate representation.

Table 3.2. The relations and corresponding codes

Relation Type	Code	Relation Type	Code
Synonym	&	Derivationally_related_form	m
PertainedBy	!	Domain_of_synset_TOPIC	n
Antonym	a	Member_of_this_domain_TOPIC	o
Hypernym	b	Domain_of_synset_REGION	p
Hyponym	c	Member_of_this_domain_REGION	q
Instance_Hypernym	d	Domain_of_synset_USAGE	r
Instance_Hyponym	e	Member_of_this_domain_USAGE	s
Member_Holonym	f	Entailment	t
Member_Meronym	g	Cause	u
Substance_Holonym	h	Also_see	v
Substance_Meronym	i	Similar_to	w
Part_Holonym	j	Verb_Group	x
Part_Meronym	k	Participle_of_verb	y
Attribute	l	Pertainym	z

During the initial phase of the algorithm, the identities of the words belonging to each word pair in real-life test datasets (MEN3000, WS353, RG65) are found and placed in a word list. All the

binary combinations of identity values in the generated word list are used to identify whether these two words are related in WordNet. All existing pairs are inserted to the training set along with the type of relationship between them. In this process, ambiguations may arise in some cases. For example, in some cases, any of the words in the word pair taken from the test dataset may have multiple nodes in the WordNet graph. For this, an disambiguation approach has been proposed.

In the stage of finding the correct nodes for words with more than one number of nodes, first all nodes of the word are found in WordNet, and a comparison is made with the help of the assumption that " in case of ambiguity, concepts that are semantically closer to each other are taken" (Pilehvar et al., 2013). Meanwhile, it is known that semantic closeness also has a positive relationship with the shortest paths between nodes. For instance, to determine the proper words represented in the given word pairs "book" and "paper" in WS353, first all nodes containing "book" and "paper" are found in WordNet. There are 13 nodes for "book" and 9 nodes for "paper." Then, to find the shortest path between nodes, a total of 13x9 = 117 word pairs are compared, and it is assumed that the one with the shortest path represents the correct nodes for the words "book" and "paper." In some cases, there may be multiple shortest paths of the same length, and in such cases, the priority status of the relationships is checked. Sblini and Kosseim (2013) have categorized semantic relationships into six different significance levels. If there exists multiple paths of the same shortest length when comparing two words, a significance level is computed for each path based on the relationship priorities outlined in Table 3.3.

Table 3.3. Relations and corresponding priority values

Relation Type	**Code**	**Relation Type**	**Code**
Synonym	1	Derivationally_related_form	3
PertainedBy	2	Domain_of_synset_TOPIC	6
Antonym	2	Member_of_this_domain_TOPIC	6
Hypernym	3	Domain_of_synset_REGION	6
Hyponym	5	Member_of_this_domain_REGION	6
Instance_Hypernym	3	Domain_of_synset_USAGE	6
Instance_Hyponym	5	Member_of_this_domain_USAGE	6
Member_Holonym	4	Entailment	2
Member_Meronym	4	Cause	2
Substance_Holonym	4	Also_see	6
Substance_Meronym	4	Similar_to	2
Part_Holonym	4	Verb_Group	2
Part_Meronym	4	Participle_of_verb	2
Attribute	6	Pertainym	2

The priority values of the relationship types are multiplied to identify the absolute relevance level of each path. The priority values of all edges from the source node to the target node are multiplied to conduct the calculation. When making a elimination, the principle of choosing the path with the lowest absolute importance is applied. If there are multiple shortest paths with the same significance level, one of them is chosen randomly.

According to earlier research, it has noted that WordNet-based approaches are influenced by two crucial factors that impact similarity. These factors are the distance of the path between two concepts and the types of relationships that form the path. Additionally, it is widely known that there is a negative correlation between path length and semantic similarity. This implies that a smaller degree of similarity is correlated with a longer path length. Although it is well known that the shortest path between concepts is a crucial factor in determining similarity, this is not enough on its own. In determining the weights of semantic relationships and calculating semantic similarity, all paths connecting two concepts must be taken into account. Since considering all paths in semantic similarity measurement has a high operational cost, the idea of considering all paths of specific lengths connecting two concepts on the WordNet graph structure has been found to be more applicable and adaptable.

Initially, it is necessary to use all paths connecting two pairs in the MEN3000 dataset. However, this would bring a huge computational cost, it is decided that all paths should have a specific length. The steps of this approach can be summarized as follows: after finding the shortest path between two concepts, all paths with a maximum length "(shortest path) + 3" are searched. "+3" is set as the maximum length to collect paths between two synsets from WordNet. This value can be chosen larger, but if this value increases, the query time increases.

The similarity score value of the word pair in MEN3000 is assigned to each path found. If there are same paths with different scores, their mode/square is taken and converted into a single similarity value. On the other side, even though the shortest path is insufficient to determine semantic similarity, it is obvious that short paths matter more than long ones in determining how similar two concepts are. Additionally, although the number of short paths between two concepts is very small, the number of long paths is very high. Even though the number of short paths is low, it is necessary to consider similarity values. Therefore, to give more importance to shorter paths, the importance factor is proposed. The aim here is to consider the importance of short paths after adding them to the word list, which contributes to having a higher similarity compared to long paths. The calculation of the importance factor is shown in Equation 3.1.

$$ImF_i\left(getAllPath\ (w_1, w_2)\right) = n^{n-k_i} \tag{3-1}$$

In Equation 3.1, "n" represents the maximum path length, and " k_i " represents the current path length. For example, let's consider the relationship "(W_1, W_2) =0.6" from the MEN3000 dataset. First, the shortest path between W_1- W_2 is found. If the shortest path is assumed to be 3, then all paths with a maximum length of 6 are added to the word list from the (3+3) formula. And all added paths are assigned a similarity score of 0.6. The data about the paths found is shown in Table 3.4.

Table 3.4. Example of importance factor

Path	Occurence	Importance Factor	Value
bcjk	6	1	0.6
jkbc	6	1	0.6
bc	1	16	0.6
jk	1	16	0.6

As seen in Table 3.4, it is detected 6 times from "bcjk" and "jkbc" paths, and 2 times from "bc" and "jk" paths. After the calculation of the importance factor according to Equation 3.1, it is observed that shorter paths have a higher importance factor.

In Table 3.5, the process of adding to the word list is continued and the situation after adding the paths of the (W_1, W_2) relationship in addition to the (W_4, W_3) relationship is shown.

Table 3.5. Example of list creation

Path	Value List
Jcjk	[0.7]
Jkbc	[0.6 0.6 0.6 0.6 0.6 0.6]
Bcjk	[0.6 0.6 0.6 0.6 0.6 0.6]
Jkjk	[0.7 0.7]
Jk	[0.6 … 0.6 (16 times) 0.7 … 0.7 (32 times)
bc	[0.6 0.6 0.6 0.6 0.6 0.6]

For each word pair in the MEN3000 dataset, the obtained paths are individually considered, and the similarity scores given to each path are kept in a separate list. The value of each path is determined by calculating the average. In Table 3.5, the path "jk" is present in both the (W_1, W_2) relationship and the (W_4, W_3) relationship. Since the importance factor for the relevant path in the (W_1, W_2) relationship is calculated as 16, a similarity score of 0.6 is assigned 16 times, and for the (W_4, W_3) relationship, the importance factor is 32, so similarity score of 0.7 is assigned 32 times. By taking the average of these values, the value of that road is determined.

In the resulting path list, where the path length is 1 (single-length paths), are the values we now obtain for the weight of each relationship. Nevertheless, not all of the WordNet relationship weights are acquired through this method. Due to the absence of single paths for certain types of semantic relationships in the MEN3000 dataset, the following method has been used to address this issue. We can identify either the shortest paths or paths containing the "x" relationship, and then acquire the weight value for the relationship type that cannot be assigned a weight using other relationship types. For example, let's assume that we have a shortest path such as “xa” with a weight of 0.5, and the known weight of 'a' is 0.7. In this case, x = 0.5 / 0.7 = 0.71 is obtained. Using the edge-counting method created by Yang and Powers (2005), a relevant weight of 0.85 is assumed if this relationship type is never observed in MEN3000. As the second step of the method, the obtained weights are applied to all paths of each word in the MEN3000 data, and the values obtained are taken as the closest approximation to the real human evaluations in MEN3000.

Placement of Prototype Positions

After determining the optimal values for the relationship weights in WordNet, the prototype of each word is determined. The optimum positions of these prototypes are found according to Euclidean-based distance calculations and the relationship weights determined in the previous step. Firstly, a random assignment process is performed to determine prototype positions for word vectors. In the next step, the randomly generated prototype positions are updated with WordNet weights. The training of prototypes is accomplished by updating the prototype positions until they get close to their relationship values. The updating of prototype positions in each training iteration is done using Equation 3.2.

$$\Delta P = \eta\ (A - B) \tag{3-2}$$

The term η in Equation 3.2 represents the learning coefficient, (A-B) represents the distance between the prototypes A and B. ΔP represents the amount, where the distance of both prototypes, the leading coefficient and the update amount are determined in the first step of the algorithm. During the training process, prototype positions are updated according to the update rate. If the similarity value calculated between the A and B prototypes ($Y_{sim(A,B)}$) is greater than the desired similarity ($D_{sim(A,B)}$), the two prototypes move away from each other; otherwise, they are moved closer to each other to achieve the desired similarity.

A distance-based similarity function is used for similarity calculation. In this work, the traditionally used similarity function based on Euclidean distance is preferred. This function is shown in Equation 3.3.

$$Y_{sim(A,B)} = \frac{1}{1 + ||A - B||} \tag{3-3}$$

In Equation 3.3, the expression ||A-B|| shows the distance between the prototypes A and B. All distance values in the work are calculated using the Euclidean distance. In Figure 3.4, it is shown that an example of moving prototypes according to 3 words and 2 relationship values what are given.

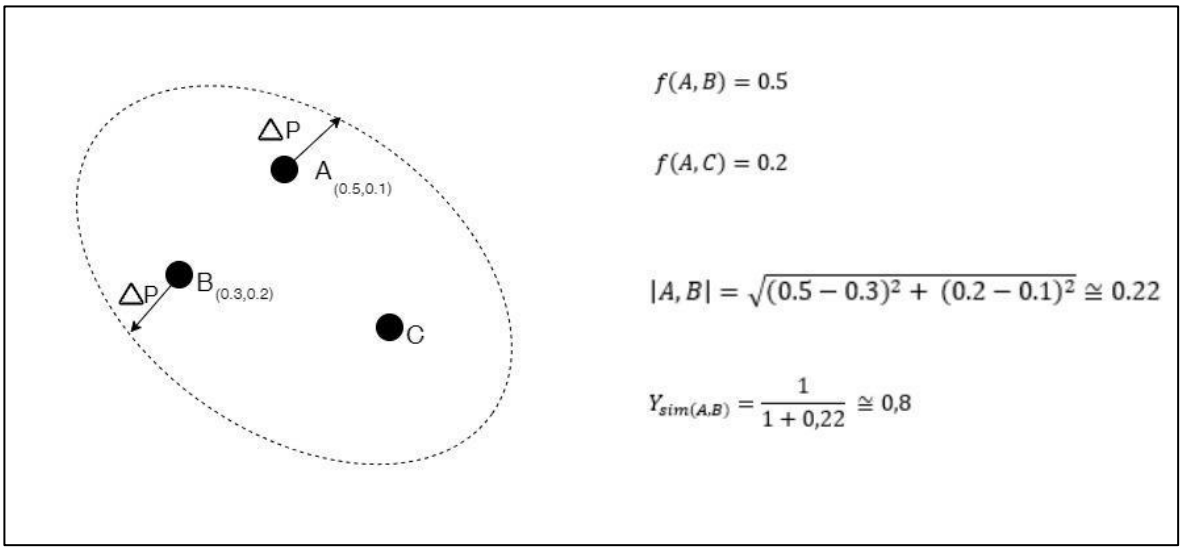

Figure 3.4. Movement example of prototypes

As seen in Figure 3.4, X, Y, and Z represent randomly placed word vectors. Since $Y_{sim(A,B)}$=0.8 is calculated according to Equation 3.3, it is observed that there is a value more closer than the value of f (A, B) = 0.5 obtained in WordNet weighting. For this reason, the distance is adjusted and the similarity is tried to be 0.5. In order to make the similarity 0.5, the required value for |A, B| is calculated as shown in Equation 3.3 below.

$$0.5 = \frac{1}{1 + ||A - B||}$$

According to Equation 3.3, |A, B| is calculated as 1. The calculation of the change amount to be used in updating the prototype positions is carried out according to Equation 3.1, and both vectors are gradually moved away from each other by the amount ΔP = η (1-0.22).

The energy values are calculated by subtracting Spearman's Correlation (SC) and Mean Absolute Error (MAE) values separately for each set, and if the E_{mean} measure values obtained from these differences are high, the positions of the prototypes and WordNet relationship weights corresponding to the highest energy value are saved. Since the aim is to discover the highest energy value, the training set (E_{tr}), test set (E_{ts}), and average (E_{mean}) energies are monitored throughout the training process, and vector positions and WordNet relationship weights that support the highest Emean value are saved. The calculation of the average energy is shown in Equation 3.4.

$$E_{mean} = \frac{SC_{tr} + SC_{ts} - MAE_{tr} - MAE_{ts}}{2} \quad (3\text{-}4)$$

The weight determination problem in WordNet requires a solution simultaneously with the optimization of prototype positions. The "success of the system on test data" is utilized as the objective function in this step, which employs a search-based methodology. This algorithm steps can be summarized as preserving the existing WordNet relationship weights and prototype positions for the situation that will lead the system to the best success. The final step of the algorithm can also be summarized as follows. New starting positions and weights are created by adding random values as α to the current best prototype positions and β to the WordNet weights. This process is repeated until the desired iteration value is reached and then the training step is performed. In other words, moving the prototypes in the algorithm according to the weights is a loop that continues to stochastically search using the learning rate (α) to get the optimal vector positions. The saved word vectors with the highest energy value are output at the conclusion of the procedure.

3.3. Proposed Method: Contextualized Deep SemSpace

In the proposed "Contextualized Deep SemSpace" approach, firstly, the generalized SemSpace method is trained with WordNet 3.1 data to determine synset vectors. Subsequently, each word within an intent dataset is transformed into a synset vector utilizing a contextualized method, and ultimately, these synset vectors are trained using a deep learning model employing BLSTM.

3.3.1. Determination of Synset Vectors

In NLP studies, the idea of converting words into vectors initially started with the "one-hot encoding" method, and this method has continued to be used for reasons such as ease of representation and independence from language. In this approach, word vectors are set to be rows of an identity matrix. In other words, the number of features in the vectors is adjusted according to the size of the vocabulary, and in each vector, only one feature is active, different from the others (Turian et al., 2010). However, this representation does not take into account the semantic relationships between words and it creates the sparsity problem that causes serious memory and computational costs (Chen et al., 2015). To address this issue, researchers have started using an approach called word embedding in their studies. The most well-known word embedding methods in the literature are Word2Vec (Mikolov et al., 2014), GloVe (Pennington et al., 2014), and FastText (Joulin et al., 2016; Bojanowski et al., 2017) approaches. In the Word2Vec method which has two approaches, while the CBOW approach aims to predict the centeral word by using neighboring words, the Skip-Gram approach aims to predict neighbors by looking at a single word in the opposite logic. Unlike these two approaches, the GloVe method calculates vectors by using corpus-based co-occurrence statistics of words in a regression equation. The FastText method is a library developed by the Facebook AI Research lab and is also effective in languages with structural differences (Bojanowski et al., 2017). Word2Vec and GloVe approaches consider words as the smallest unit to work with and are similar to each other in this respect. FastText, on the other hand, approaches each word in a

structure consisting of character n-grams. Despite their success in NLP studies, all of these word embedding approaches have an important limitation because they represent each word as a single vector, even if they have different meanings. In order to eliminate this limitation, an approach called sense embedding has been proposed in the literature.

Sense embedding learns the representation vectors of words by using their neighborhood in a corpus, similar to the word embedding approach, but with a focus on their meanings as well. Sense embedding methods are generally divided into two categories: unsupervised (Brody and M. Lapata, 2009; Li and Jurafsky, 2015; Pantel and Lin, 2002) and knowledge-based (Chen et al., 2014; Camacho-Collados et al., 2016; Sinoara et al., 2019; Jauhar et al., 2015; Rothe and Schütze, 2015). In the unsupervised learning approach, word meanings are learned by looking at their properties in the corpus, while the knowledge-based approach a vocabulary such as WordNet is used as the main source. In this way, both existing information in the corpus and semantic relationships encoded in the vocabularies are used. Although it has a more logical basis than word embedding methods, unfortunately the best results in the literature do not belong to sense embedding methods. In a different sense embedding method study, Orhan and Tulu (2021) generated word vectors in Euclidean space by weighting only the semantic relationships found in WordNet without using any corpus. During the optimization process in the study, the semantic weights in WordNet are aligned to datasets in which the semantic relationship values are scored by humans. Thus, it is determined what weight each semantic relationship corresponds to between [0 and 1] in people's minds. Additionally, the calculated word vectors are uniquely identified both semantically and lexically. In other words, a separate vector is identified for each word in a synset. Unlike the Orhan and Tulu (2021) study, this study used a single vector representation for each synset defined in WordNet. Because in the Semspace algorithm, it is observed that all weights remained constant at approximately 0.5. This finding shows that the SemSpace technique only considers the existence of a semantic relationship between any two words, regardless of their relationship type. Additionally, it is not always easy to find datasets such as MEN3000, which is used for alignment in the SemSpace algorithm. For that reason, in our approach a random search process (without alignment) is applied for WordNet relationship weights. A dataset representing every synset in WordNet 3.1 and every relationship between them is created for this aim, and the SemSpace (Orhan and Tulu, 2021) algorithm is run to determine each synset's vector.

There are 117,791 synsets, 365,083 semantic relations and 26 different types of semantic relation in the Wordnet 3.1 (Fellbaum, 2012) data provided. With the aim of finding a vector for each synset, a dataset containing all semantic relationships in the format (Synset1, Synset2, RelationType) is prepared. Following that, the generalized SemSpace algorithm is executed with the aim of concurrently determining both the weights of the semantic relationships and the vectors associated with the synsets. In this dual optimization problem, the weight randomly assigned to the relationship defined between both synsets in WordNet is aimed to be the same as the distance-based similarity in

the Euclidean space of the vectors representing these two synsets. Values (both vectors and weights) where the difference of the cost function "Sperman Correlation - Mean Absolute Error" is maximum are stored. While weights are randomly searched, vectors are moved closer and further away from each other in order to maximize this cost. These calculations begin with both vectors and weights being randomly assigned. Then, according to the relationship dataset, the similarity values between vectors are calculated using Equation (3.5).

$$Sim(V_1, V_2) = e^{-\|V_1 - V_2\|} \tag{3-5}$$

Here, the terms V_i represent the synset vectors that are defined as having a relationship between them. The calculated similarity value, which is in the [0, 1] range, is compared with the randomly searched relationship weights also in the same range. When the calculated similarity value (Sim) is greater than the expected relationship weight (W), the vectors move away from each other; otherwise, they move closer. This updating movement of vectors is shown graphically in Figure 3.5.

Figure 3.5. Training synset vectors in a Euclidean space

The Sim function shown in Figure 3.5 represents the similarity value calculated by equation 3.5, and the W term represents the weight determined by random search for the semantic relationship between the synsets of the two vectors in WordNet. In Figure 3.5 (a), it can be seen that the two vectors are too close to each other and are moved away from each other. On the contrary, in Figure 3.5 (b), vectors that are far away from each other are brought closer. These two movements form the basis of generalized SemSpace training and is shown in Algorithm 1.

Algortihm 1 *Training Algorithm*

Input: *List to Train* ***T****, Training iteration count* ***N,*** *Dimensions of semantic space* ***D****, MaxRepeatIter*
Output: *V, vectors*

```
For(1) each synset S in V
    S.vector ← random(D)
End For (1)
For(2) each relations Relation in V
    Relation.weight ← random([0,1])
End For (2)
For(3) (i=1 to MaxRepeatIter)
    While (error > Threshold AND i < N)
        For(4) each synsetPair (S1, S2) in V
            X, Y ← Find_synset_vectors_in_Vocab(S1,S2)
            UpdateSynsetVectors(X,Y,R(S1,S2))
        End For(4)
        E ← ComputeEnergy(T,R)
        if (E > MaximumEnergy)
            MaximumEnergy ← E
            saveChanges()
        End if
        For(5) each synset S in V
            S.vector ← α * (0.5 - random(D))
        End For(5)
    End While
End For(3)
return V vectors
```

The WordNet synset node relationships are listed in the TS, training set. All of the synsets in the vocabulary list are given random vector positions at the beginning, and all of the relations are given random values between [0,1]. Once the highest number of iterations had been reached, the learning loop stopped. Each synset vector is taught by training iterations inside of this loop until a predetermined number of times (N) or until the mean absolute error (error) threshold value. The equation for Compute_Energy(TS, R) is provided in (3.4). Then, the vectors with highest energy are kept. In order to update the vector positions, UpdateSynsetVectors(X, Y, R(S1, S2)) computes the Euclidean distance-based similarity between the X and Y vectors, by comparing the computed similarity and R(S1,S2) values. The quantity of updates to vector positions is determined by equation 3.6.

$$\Delta = \eta\,(V_1 - V_2) \tag{3-6}$$

Here, the term Δ represents the amount of update in the positions of vectors, the η term represents the learning rate, and the V_1 and V_2 terms refer to the vectors that are related to each other. To ensure the consistency of the vectors to be obtained after training, there should be a sufficient number of relationships in the dataset. If every relationship between two synsets is thought as a two-

variable equation, the inconsistency problem that might come up when using two equations to determine the positions of three vectors, in other words the "lack of equation problem" (Tulu, 2019; Orhan and Arslan, 2020), is shown in Figure 3.6.

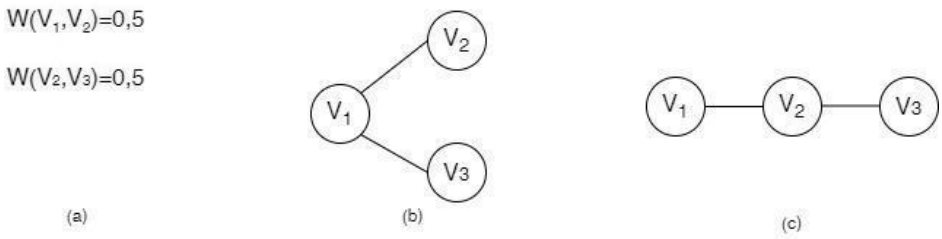

Figure 3.6. The problem of lack of equations in generalized SemSpace training

In Figure 3.6.a, it can be observed that two equations are given for three vectors. Traditionally, it is known that there are infinitely many solutions for three unknowns and two equations. Here, only two solutions are presented. Although Figure 3.6 is not a graph representation, an edge is drawn between the vectors to indicate that a relationship is defined between them. With the generalized SemSpace approach, although it is guaranteed that these three vectors will be close to each other, the exact relative positions of V_1 and V_3 are not fully controlled. Thus, the training can end as in Figure 3.6.b or Figure 3.6.c. It is clear that the calculated Sim(V_1, V_3) value in Figure 3.6.b will be much larger than the Sim(V_1, V_3) value calculated in Figure 3.6.c. This problem, known as the "equation deficiency," is one of the weaknesses of the generalized SemSpace approach that can create a representation problem. Another weakness is its inability to find vectors for words outside the training dataset, referred to as Out Of Vocabulary (OOV). However, OOV is a very common issue with all word embedding techniques. This problem will be discussed in more detail in the analysis section.

In the detailed analyses, it has been revealed that in smaller vector dimensions (e.g., 3-dimensional space), unrelated synsets unintentionally become neighbors, while in larger dimensions (e.g., 300-dimensional space), more semantically correct neighborhoods are formed. This situation can be interpreted as the fact that in high-dimensional vector training, vectors trained only by "get closer" do not need to "move away" and therefore come to more accurate positions (Orhan and Arslan, 2020). In the literature, it is seen that word vector representations of up to 1000 sizes and more have been produced for large language models, which are recent studies (El-Alami et al., 2022; Seker et al., 2022; Seo et al., 2022; Seyyar et al., 2022; Zhang et al., 2022). However, since these are sentence-level analyses, there is no need to produce such high-dimensional vectors in our own study. Therefore, in this study, depending on the memory limits of the graphics card used, vectors of up to 300 dimensions could be used.

However, in WordNet, some words can be found in different synsets. For example, the word "cheap" has 4 different synsets in WordNet. Deciding which synset vector will represent a word requires serious word sense disambiguation solution. The disambiguation of which synset a word in any given text should be considered under is clarified using the sense disambiguation approach described in the following section.

3.3.2. Word Sense Disambiguation

Word Sense Disambiguation (WSD) is one of the most challenging tasks in artificial intelligence and is considered as an AI-complete problem. WSD, whose main application areas are machine translation, information retrieval, knowledge extraction and text mining, is actually used in almost every kind of language research (Pal and Saha, 2015). Originally conceived in the late 1940s as a core task of machine translation (MT), over time it has become clear that WSD is a highly complex problem (Navigli, 2009). It also plays a vey important role in improving the quality of NLP systems. In this study, sense disambiguation is considered not only one of the encountered challenges but also one of the most important stages of the research.

In NLP, WSD is the process of determining the correct senses of words with multiple senses in the context using various computational methods (Navigli, 2009). The need for WSD arises from the fact that many words have more than one senses and determining the intended sense is essential for the accurate understanding and processing of natural language. The difficulty of WSD stems from the fact that many words in natural language can be interpreted in many ways depending on the context they are in. For example, consider the following two sentences.

a) I need to book a restaurant for our anniversary.
b) I like to read adventure books.

In the first sentence, the word 'book' refers to making a reservation, while in the second sentence it means printed work consisting of pages.

WSD can actually be thought of as a classification problem in which word senses are assumed to be classes. It is the use of an automated classification method to assign the occurrence of each word to one or more classes, based on data from context and knowledge sources. (Navigli, 2009). Unlike other classification tasks in the field of NLP (such as part-of-speech tagging, named entity resolution, and text categorization), WSD doesn't use a predefined single set of classes. Instead, the class set varies depending on the word to be classified (Navigli, 2009).

WSD relies heavily on knowledge. In fact, the core procedure of any WSD system can be summarized as follows: when given a set of words, a technique is applied to associate the most suitable senses with the words in context, using one or more sources of knowledge. These knowledge sources can vary significantly, from text corpora that are untagged or annotated with word senses, to

more structured sources such as machine-readable words, semantic networks, etc. (Navigli, 2009). In the literature, different approaches have been proposed for the WSD process, and many studies have been conducted. To build WSD models, various rule-based methods or machine learning approaches are used. Machine learning models use algorithms such as Support Vector Machines (SVMs), Decision Trees or deep learning models such as RNNs or transformer-based models (BERT) to make predictions. Early proposed knowledge-based approaches for WSD have been replaced by more powerful machine learning and statistical techniques in last few years. Today, we can classify WSD methods under three major categories: knowledge-based, supervised, and unsupervised.

Knowledge Based Approaches: These are approaches that have emerged since the 1970s and use various vocabularies or synsets sources such as WordNet, SemCor, Wikipedia to find the appropriate senses of a word in a given context. Knowledge-based approaches use either hand-encoded rules and grammatical rules or the sense that most closely resembles the context of the ambiguous word (Chandra, 2014). In knowledge-based approaches, many methods are used.

One of the most common approaches is the overlap-based algorithms which derived from the LESK algorithm (Lesk, 1986), which is a machine-readable vocabulary-based algorithm. The Lesk algorithm defines the correct sense of a word by comparing the context with the vocabulary definitions of candidate senses. This algorithm carries out WSD by looking at the overlap of vocabulary definitions/descriptions of words in a sentence. Firstly, a short phrase is selected from the sentence containing the ambiguous word. Then, definitions for different meanings of the ambiguous word and other meaningful words found in the phrase are collected from an online vocabulary. Afterwards, all definitions of the target word are compared with the definitions of the other words. The sense with the maximum number of overlap is considered to represent the intended sense of the ambiguous word. For example, in the sentence "time flies like an arrow," the algorithm compares the definitions of "time" with all the definitions of "fly" and "arrow." This approach has two disadvantages. The first one, as the number of words in the text increases, the complexity also increases since the number of comparisons is directly proportional to the number of words in the text. The second disadvantage is the expressiveness of the definitions is limited because the overlap is based solely on co-occurrence of words (Pal and Saha, 2015).

In subsequent research, the Lesk algorithm has been extended and adapted. Banerjee and Pedersen (2002) proposed the "Extended Lesk" algorithm in which they also used WordNet, in their research. They have included synonym, hypernym, and hyponym from WordNet to improve the accuracy of the disambiguation. Similarly, Agirre and Edmonds (2006) proposed a similarity-based approach using WordNet in their studies. Navigli et al. (2005) introduced a WSD approach based on the structural pattern recognition framework. Basile et al. (2014) proposed a method based on the use of a word similarity function defined in a distributional semantic space to calculate description-

context overlap. In their method, which they presented as a variation of the Lesk algorithm, they used BabelNet, a large multilingual semantic network created by leveraging both WordNet and Wikipedia as the semantic inventory.

Another category of methods is based on semantic similarity. These methods rely on the assumption that semantically related words share a common context, and the appropriate sense of an ambiguous word can be determined by these associated senses. Several similarity measures are used to identify how semantically related two words are.

Supervised Approaches: These are approaches based on the assumption that context alone provides sufficient evidence to disambiguate words. In these approaches, models are trained using a labeled dataset to disambiguate word senses. The most commonly used methods in supervised approaches can be listed as decision trees and SVMs. Furthermore, recent advancements in deep learning have led to the development of NN-based WSD models. These models typically use word embeddings, context representations, and attention mechanisms to resolve ambiguity.

One of the first effective works in supervised WSD is conducted by Yarowsky (1995). In this study, the idea of using a small amount of labeled data and a large amount of unlabeled data to perform WSD has been explored. Ng and Lee (1996) also proposed a SVM-based approach for WSD in their work. They combined different sources of knowledge, such as WordNet and choice-based preferences to disambiguate words with ambiguous senses. Iacobacci et al. (2016) introduced "Sense2Vec" in their research. Sense2Vec is a NN-based approach that learns word sense embeddings from a large corpus and performs WSD using the cosine similarity between context and sense embeddings.

Unsupervised Approaches: These are methods that are not dependent on external knowledge sources, machine-readable vocabularies or semantically annotated datasets (Pal and Saha, 2015). Starting from the assumption that similar sense will be found in similar contexts, it focuses on the fact that ambiguous meaning can be extracted from the text by clustering word occurrences using a measure of context similarity. Representing of words with fixed-dimensional dense vectors using word embedding techniques has become one of the foundations of many NLP systems. Lexical databases such as WordNet and BabelNet are also used with unsupervised approaches in addition to word embedding methods. The main approaches in unsupervised methods can be classified as context clustering, word clustering, co-occurrence graph, and spanning tree-based approach. Context clustering methods are based on clustering techniques in which context vectors are first generated and then they are grouped into clusters to determine the sense of the word. (Niu et al., 2004). This method uses a vector space as the word space and its dimensions are just words. Again in this method, a word in a corpus is represented as a vector and the number of times it appears in context is counted. Then, a co-occurrence matrix is constructed and similarity measures are applied. Subsequently, any

clustering technique is used to make distinctions. Word clustering is similar to context clustering in terms of finding senses, but it clusters words that are semantically identical. The co-occurrence graph method uses graph-based techniques to represent word relationships and determine senses and creates a conflict graph. These methods create graphs where words are nodes, and edges represent semantic relationships. If words are found together in a syntactic relationship in the same paragraph or text based, an edge (€) is added.

Mihalcea (2002) proposed the idea of using PageRank-like algorithms for WSD. In this study, it is presented an unsupervised method to assign the correct senses of words by taking the advantage of the structure of WordNet.

Although the methods proposed for WSD have made significant progress, improving the accuracy and performance of disambiguation methods continues to be an active research area in NLP. Various methods are being explored to achieve better results in WSD, including deep learning-based approaches, transformer models and large-scale pretrained language models.

In the WSD stage of our study, it is aimed to find the correct synsets corresponding to the words in the text in the datasets. First, standard preprocessing steps of NLP studies (such as filtering stop-words, lemmatization, and tokenization) are applied to the datasets. As an example, the preprocessing steps for the sentence "what flights from Salt Lake City to New York City arrive next Saturday before 6 pm" are shown in Table 3.6.

Table 3.6. The data preparation steps

Step	Output
Original Text	what flights from salt lake city to new york city arrive next Saturday before 6 pm
Filter Stop-words	flights salt lake city new york city arrive next Saturday 6 pm
Lemmatization	flight salt lake city new york city arrive next Saturday 6 pm
Tokenization	flight-salt lake city-new york city-arrive-next-Saturday-6-pm

After completing the procedures in Table 3.6, each token's potential synset candidates have been identified. Some tokens may have only one synset candidate, while others may have more than one synset candidates. OOV words not defined in WordNet have been ignored and have not been included in the rest of the analysis process. Table 3.7 shows all the synset candidates found for the tokens obtained from the final output in Table 3.6.

Table 3.7. Suitable synset candidates for example setence

Lemma	**SynsetID**
Flight	108237455, 100303220, 103367905, 100059563, 108237361, 105632993, 111501734, 108237541, 100302018, 202490360, 201944727, 201673687
Salt Lake city	109170480
New York city	109141944
arrive	202009962, 202591952
next	300128838, 300449506, 301298098, 300129027, 400054750
Saturday	115189617
6	113766862, 302194472
pm	106292501, 100142216, 114675658, 109926654

Table 3.7 demonstrates that whereas a general-purpose term like "flight" has 12 synset candidates, name entity tokens like "salt lake city" and "new york city" only have one. For words with more than one synset candidates, the disambiguation process, in other words, finding the appropriate sense of the word in the sentence, is carried out. Although there are many different methods in the literature, researchers commonly agree on conducting contextual analysis to resolve ambiguity of word sense (Eryiğit, 2012).

In this study, in order to apply context analysis to the intent detection problem from a different perspective, a different approach that performs calculations in Euclidean space, where generalized SemSpace vectors are defined is used. It has been observed that there is an intuitive context cluster in the intent detection datasets within the same topic framework. As a result, in the suggested WSD technique, each dataset's context is first represented by the N words (N = 5 in this study) that appear the most frequently. This N value may be raised or lowered depending on the amount of synset candidates of commonly used words in that dataset, taking the computational cost into account. The most frequently used and non-OOV words are identified, and every synset candidates corresponding to these words are investigated. These selected words actually form the context cluster that summarizes the dataset studied. For example, by considering the entire ATIS dataset, the top 5 most frequent lemmas are selected, and all synset candidates for these lemmas in WordNet are examined. Table 3.8 shows the 5 lemmas identified and the IDs of synset candidates for each lemma.

Table 3.8. The most frequent words in the ATIS dataset and IDs of their synset candidates

Lemma	SynsetID
flight	108237455, 100303220, 103367905, 100059563, 108237361, 105632993, 111501734, 108237541, 100302018, 202490360, 201944727, 201673687
boston	109118343
show	100521313, 106892571, 106631572, 100756620, 202153218, 200666706, 201017253, 202141597, 201690851, 200945869, 200925764, 202144017, 200924838, 200925278, 202004579, 201088960
denver	109090592
San francisco	109088034

In Table 3.8, the first column indicates the lemma and the second column displays the ID values of all synset candidates defined in the network for each lemma for the ATIS dataset. It is assumed that the context cluster prototypes are close to each other in the generalized SemSpace space. In this context, to determine the correct prototypes, all possible combinatorial sequences are generated among these synset candidates. For the example given in Table 3.8, the representation of all formations is given in Figure 3.7.

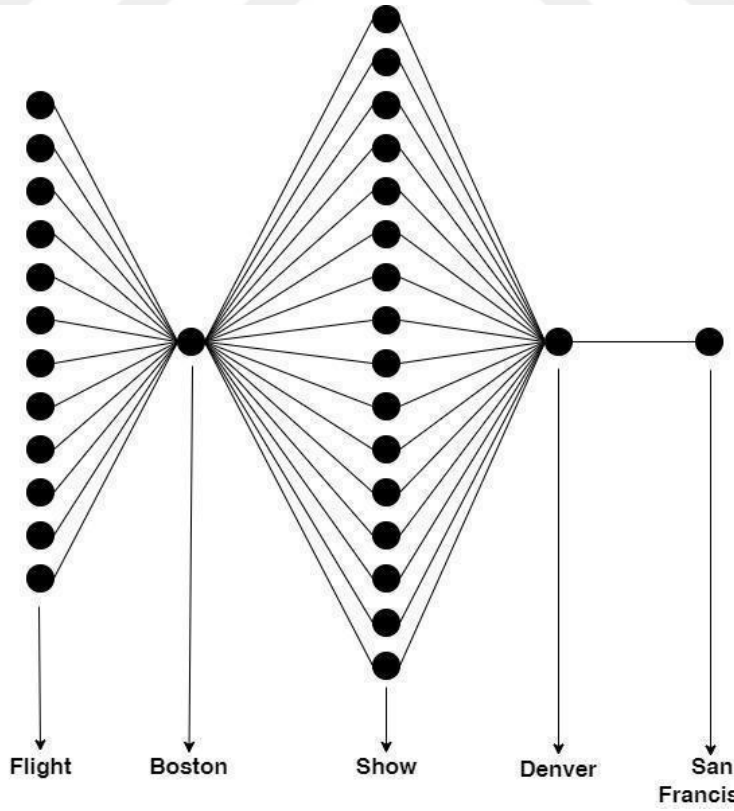

Figure 3.7. All possible sequences of the ATIS dataset`s context cluster candidates

As seen in Figure 3.7, since the lemma "flight" has 12 different synsets, it is represented by 12 nodes, the lemma "show" represented by 16 nodes, and the lemmas "Boston, Denver, San Francisco" are each represented by one node. According to the basic assumption, these prototypes are assumed to be members of a context cluster that are close to each other in space, and therefore, the order of lemmas in the sequence is considered unimportant. The distances between neighboring synset vectors in the sequence are calculated using the generalized SemSpace vector of each synset candidate, and the sum of all distances in the sequence is used to determine the total cost of that sequence. According to the proposed WSD approach, it is assumed that the combinatorial sequence with the lowest cost contains the context cluster prototypes. The disambiguation process of the words that appeared in the dataset is carried out by selecting the synset candidate closest to the context cluster prototypes. An example of the distance calculations between the "next" lemma with 4 different synset candidates in WordNet and the 5 prototypes of the context cluster is shown in Figure 3.8.

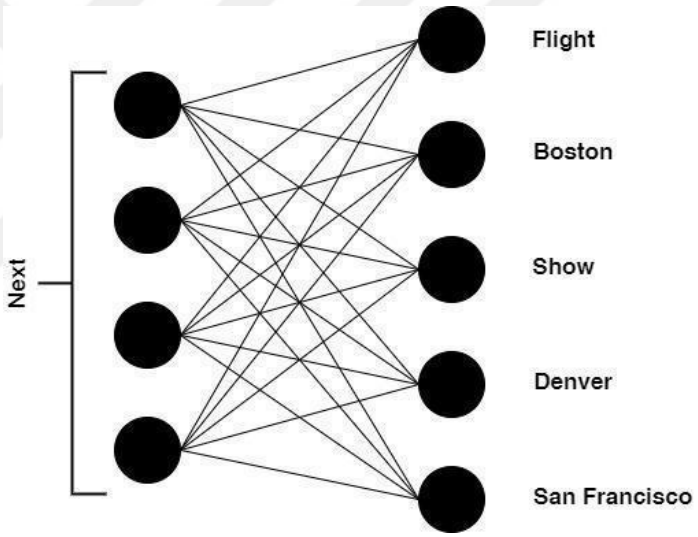

Figure 3.8. An illustration of finding the “next” synset candidate that is closest to context cluster

Since the word "next" has 4 different synset candidates, it is represented by 4 nodes in Figure 3.8, and the synsets in the prototype, which are initially determined according to the most frequently occurring lemmas in the data set, are also represented by a node each. For each of the synsets belonging to the word "next," the distances to the synsets of the 5 lemmas in the prototype are calculated and summed, and the synset of the "next" lemma in the combination with the shortest distance is selected as the appropriate synset. In this way, the disambiguation process of the word "next" has been performed. The process of selecting the appropriate synset based on context with the WSD approach used is given in Equation 3.7.

$$S_{WSD} = argmin \sum_{i=1}^{N} \|C_j - P_i\| \quad (3\text{-}7)$$

In Equation 3.7, S_{WSD} is the synset vector to be determined, N is the number of prototypes, C_j is the j[th] synset candidate vector of the word to which the disambiguation will be applied, and P_i is the vector of the i[th] prototype.

3.3.3. Building Deep Learning Framework

The primary purpose of the NLU module in dialogue systems is to label a user's request with context-specific intents of interest, in other words, to classify it. The classification process, which has become a very easy problem thanks to modern machine learning approaches, accepts numerical inputs for high performance. Nowadays, approaches known as word embeddings are used for the numerical representations of texts, and very successful results are obtained. Therefore, it can be said that in innovative approaches, the intent detection problem is addressed in two steps: numerical representation of texts and classification.

As in most NLP fields, probabilistic methods have been abandoned in intent detection studies. In intention detection studies in the literature, probability-based systems are used only for voice data (Korpusik and Glass, 2017; Raymond and Riccardi, 2007). In almost all recent studies, it can be said that the classification infrastructure is based on RNN architectures (Yao et al., 2014; Zhang and Wang, 2016). In these studies with RNN architecture, it is clear that the most popular method is BLSTM. Although RNNs theoretically have the capability to capture long-range dependencies, due to the vanishing gradients problem, it becomes difficult for this architecture to learn long-range correlations (Bengio et al., 1994). The vanishing gradients problem describes the situation where some parts of the weights in the RNN tend to stop changing during the learning process. Gradient values will diminish and disappear exponentially in backward propagation. As a result, prioritizing current information can lead to the neglect of past events. Therefore, repeated relationships over a long period (long-term dependencies) cannot be learned sufficiently. LSTM networks (Hochreiter and Schmidhuber, 1997) are designed to overcome these problems and are capable of handling long-term dependencies in the text. An LSTM layer consists of a repetitively connected sequence of blocks known as memory cells (Graves and Schmidhuber, 2005). It includes a gate mechanism that controls the recursive addition and deletion of information from a propagating cell state in order to control the entire flow of information within neurons. Thus, the forgetting process can be controlled, and a defined memory behavior can be realized to model both short-term and long-term dependencies. The general architectural structure of LSTM is shown in Figure 3.9.

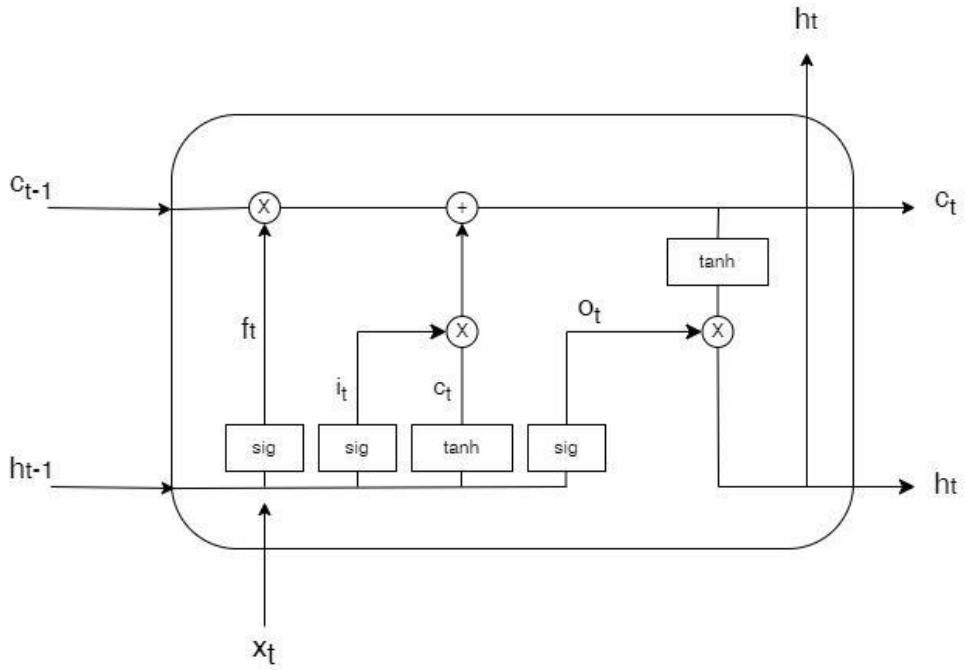

Figure 3.9. An LSTM cell

Each LSTM cell, referred to as a memory cell, consists of structures called input-forget-output gates which control the flow of information. Similarly, an output gate unit is used that protects the irrelevant memory content stored in a memory cell and other units from corruption. The forget gate scales the internal state of the cell before adding the information to the cell as input via its self-repetitive connection and adaptively forgets or resets the cell's memory (Gers and et al., 2002). The information from the previous cell (h_t) and the current information (x_t) is passed through a sigmoid activation function and a decision is made based on the result. Depending on the outcome of the sigmoid operation, the input gate determines whether to update the previous and current information. Additionally, the tan_h activation function is used to compress the data into the range of -1 to 1 for data scaling. The output gate determines the input of the next cell (h_t+1). The calculations for the input, forget and output gates are shown in Equations 3.8, 3.9, and 3.10.

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{3-8}$$

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{3-9}$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{3-10}$$

While one directional LSTM preserves only information from the past, using this structure bidirectionally operates the inputs in two ways, one from the past to the future and the other from the future to the past. BLSTM utulizes sequential learning in both forward and backward directions in two layers, from previous and future context based on the current location. To create a BLSTM network, connections between layers are made so that the output of each hidden layer (composed of both forward and backward LSTM layers) is propagated to both the forward and backward LSTM layers that constitute the consecutive hidden layer. The architecture of this design is illustrated in Figure 3.10.

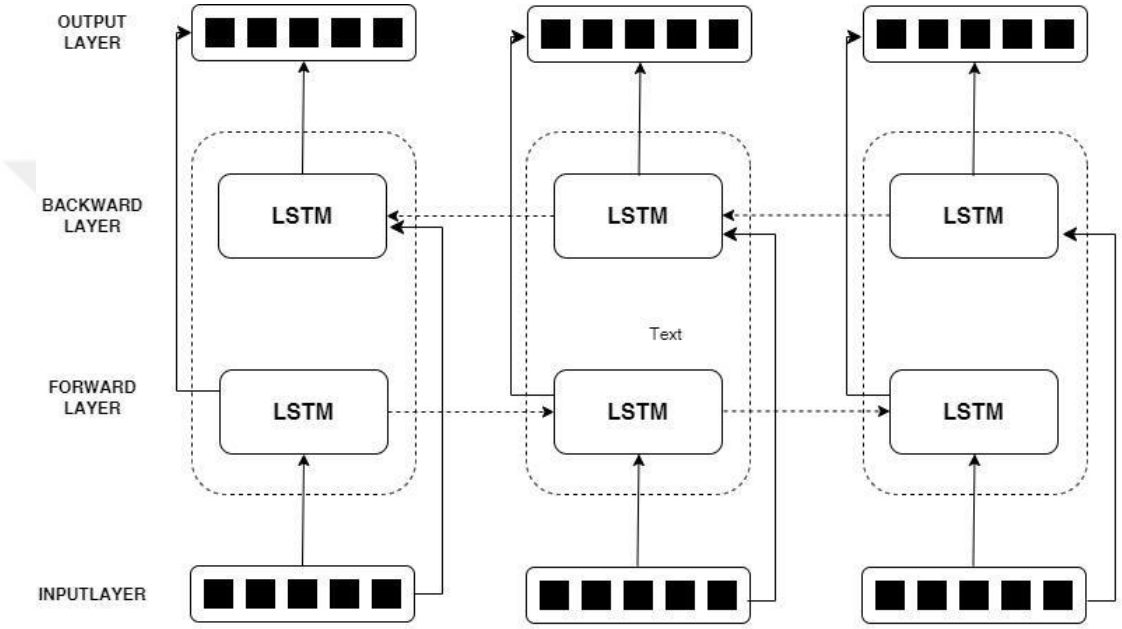

Figure 3.10. BLSTM architecture

As shown in Figure 3.10, BLSTM is an improved version of the LSTM architecture and consists of two separate LSTM layers. In BLSTM, the input data sequence (x_{t-1}, x_t, x_{t+1}) is used as input in both the forward LSTM and the backward LSTM. At each time step, both h_t (forward LSTM hidden state) and h_t (backward LSTM hidden state) are computed simultaneously.

Due of the BLSTM network model's success in the intent detection literature, deep learning techniques are our main emphasis in the study's classification phase (Kim et al., 2016; Tur et al., 2016). Although there are many studies referred to as BLSTM in the literature, most of them use the BLSTM network in together with "different layer models recommended in deep learning architectures". In this study, BLSTM architecture is preferred from a similar perspective. Table 3.9 presents the model's layer architecture in sequentially.

Table 3.9. The layer architecture of the model used

No	Layer	Detail
1	Embedded	Synset Vectors
2	BLSTM	60 neurons
3	Pooling	General Maximum
4	Drop out	Reduces neurons into %10
5	Fully Connected	50 neurons and ReLU activation
6	Drop out	Reducesneurons into %10
7	Classify	Softmax Activation

As seen in Table 3.9, the study uses a BLSTM architecture composed of multiple layers. These layers essentially carry out the functions of feature extraction, classification, word embedding, and learning sequential word patterns. The BLSTM architecture's output is planned to select one of the dataset's intents ReLU activations are used in our model's intermediate layers, while softmax activations are used in the output layer. Dropout is a very effective regularization technique to prevent overfitting of the network. We use dropout for regularization and the "Adam" optimizer for optimization and the learning coefficient $\beta1 = 0.9$, $\beta2 = 0.999$ and learning rate = 0.001 were chosen. Model parameters are updated using categorical cross-entropy.

The general architecture of the proposed model is shown in Figure 3.11.

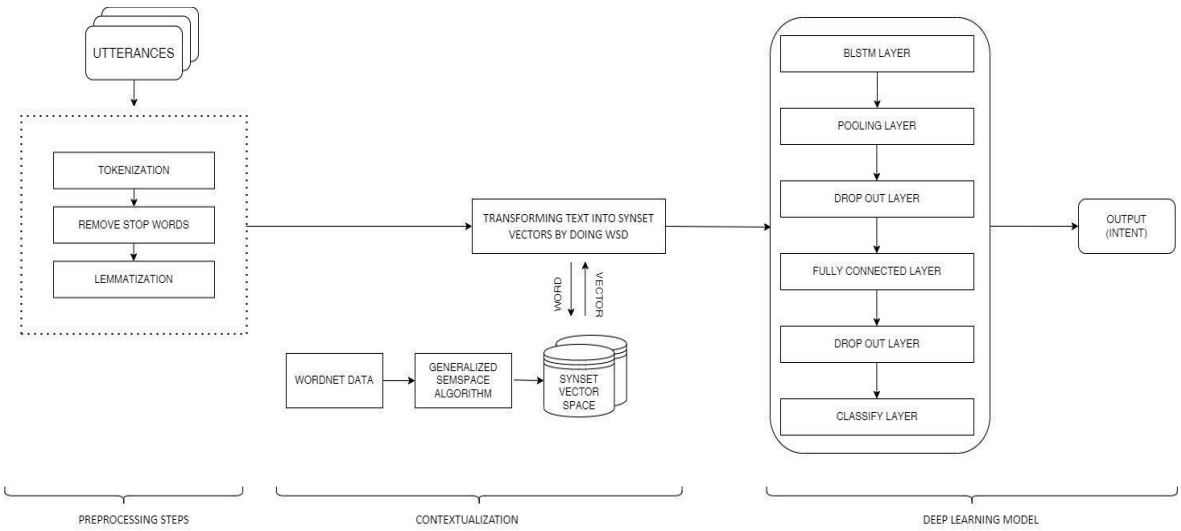

Figure 3.11. General architecture of contextualized Deep SemSpace

The contextualization procedure is carried out in this study in two parts. As the first phase, the disambiguation technique is used to determine the synset vectors based on the context cluster of the datasets. The second step is to feed the vector sequences of sentences created according to the disambiguated word vectors to the BLSTM network. Each utterance in each dataset is entered into the BLSTM network to detect intentions. Each word is initially represented by a 300-dimensional vector, and an input expression $x = (w_1, w_2, ..., w_n)$ consisting of n words is given to the model. If a word cannot be matched with any synset in WordNet, it is regarded as OOV and is excluded from the procedure. In the suggested system, preprocessing steps are applied before feeding into the BLSTM network. The data cleaning process is divided into four steps. First, words are tokenized and stopwords are removed. Then, the tokens are subjected to lemmatization, and their corresponding values on synset vectors are found. Some tokens have only unique synset candidates, while others may have more than one synset candidates. In the case of multiple synset candidates for the relevant word, the WSD procedure is applied as described in chapter 2.3.

4. RESULTS AND DISCUSSIONS

To assess the effectiveness of the suggested approach, ATIS, Facebook's multilingual, WebApp, Snips, AskUbuntu, and Chatbot datasets, which are widely known in the intent detection literature and used by many researchers are preferred. The words in these datasets are converted into synset vectors with the proposed generalized SemSpace method in this study. Synset vectors calculation is based on WordNet 3.1 data. During the vector calculation, the weights of the semantic relationships defined in WordNet are randomly initialized in the [0, 1] range and modified in random small amounts to maximize the performance of generalized SemSpace. As a result of multiple repeated trainings, it is observed that all weights remained constant at the 0.5 level. This finding suggests that the generalized SemSpace technique only considers the existence of a semantic relationship between any two synsets, independent of the types of relationship. Therefore, when defining a new semantic relationship (disregarding its type), it can be interpreted that adding this relationship as a new row to the generalized SemSpace training set may be sufficient.

Words can carry different senses and be interpreted in different ways depending on the context within a sentence. In the field of NLP, WSD, which is determining the sense of a word in a certain context, plays an important role in increasing the accuracy of studies. In our study, finding the most appropriate sense of the words in the context is of great importance in achieving intent detection with high success. While decisions for WSD are usually made by looking at context features such as neighboring words, in this study, we chose to determine the words that summarize the data set by examining the entire data set we tested. For this purpose, we selected the five most frequently occurring lemmas in the dataset and performed the WSD process by minimizing the sum of the distances of the ambiguous words to these lemmas. In this way, we have significantly overcome the problem that different senses of the same word may be closely related to each other, which is an important problem encountered in the WSD process.

4.1. Analysis of Out-of-Vocabulary

OOV word is one of the primary causes of performance declines in NLP studies (Daumé and Jagarlamudi, 2011). One of the issues affecting the results of this study is the presence of words called OOV, which are not included in the training dataset but appear in the test data. OOV typically represents a problem that includes special concepts such as proper nouns, abbreviations, place names and significantly affecting the success of NLP studies. The OOV problem is a common problem in the field of NLP and arises when a machine learning model encounters words or signs that are not seen during the training phase. The OOV problem can be particularly challenging for traditional NLP models based on n-grams or fixed-size words. Because these models are limited to the words they are trained on. When a model encounters an OOV word, it cannot provide meaningful predictions or

representations for that word, which can lead to potential errors and a decrease in performance (Lochter et al., 2020).

In this study, the presence of special terms like names, abbreviations, and locations that are not found in WordNet, causes to the OOV problem to be a frequently encountered issue. The OOV words in the study are not included in the generalized SemSpace definitions. Therefore, it is not possible to convert these OOV words into synset vectors and use them in these calculations. However, it has been observed that some sentences are represented by only a few (or even none for some) words because they contain too many OOV words. Determining intent based only on this limited number of words may be insufficient in some cases. Furthermore, it is inevitable that the increase in the number of OOV words in sentences will negatively affect the classification performance of the contextualized Deep SemSpace model. In Table 4.1, two example sentences from the WebApp dataset are shown to illustrate the OOV problem.

Table 4.1. Two sentences of the WebApp dataset

Dataset Sentence	Remaining Lemmas	Intent
How do I delete my facebook account	Delete account	Delete account
Photo sharing site (alternative to flickr) supporting openid signin	Photo sharing site support sign	Find alternative

As a result of feature extraction, it is observed that while the number of words in certain sentences decreased significantly, the number of words in other sentences did not change significantly. In the first example given in Table 4.1 since the words related to intent detection are not removed, the OOV word problem in this example is not affect the classification result. However, in the second example, since the word "alternative" is treated as an OOV word, the classification resulted in an incorrect prediction. In addition, due to the OOV problem, it is seen that the classification method generates purely statistical answers for some sentences that are not contain words. When the literature is examined, it is seen that OOV word rates are used in comparisons to affect performance in open-domain systems, although not in closed-domain systems (Samin et al., 2022; Tong et al., 2022). Based on the OOV indexes in the literature, some new indexes, including OOV_{mean} and OOV_{weak}, are defined and used to measure the effect of OOV problem on the success of solving the intent detection problem. And, some statistics for the datasets utilized to evaluate the impact of the OOV problem on the success of the proposed model are presented in Table 4.2.

Table 4.2. Some statistics of the datasets used

Dataset	Attribute					
	#Samples	#Classes	OOV_{mean}	OOV_0	OOV_1	OOV_{weak}
Atis	5.871	26	**5.7**	25	63	**1.5**
Snips	13.784	7	**4.7**	13	272	**2.1**
Facebook	38.841	12	**3.5**	61	1039	**2.8**
WebApp	83	7	**3.5**	0	6	**7.2**
Ask Ubuntu	153	4	**5.6**	8	19	**17.6**
Chatbot	205	2	**1.7**	17	75	**44.8**

In Table 4.2, '#Samples' represents the total number of samples in the datasets, while '#Classes' represents the number of intents in each dataset. OOV_0 indicates the number of samples that do not contain any words due to OOV, OOV_1 indicates the number of samples containing only one word, and OOV_{mean} is used to express the average number of words remaining in the datasets. At the bottom row of the table, the measurement called OOV_{weak} is calculated using Equation 4.1.

$$OOV_{weak} = 100 * \frac{OOV_0 + OOV_1}{\#Samples} \tag{4-1}$$

OOV_{weak} shows the proportion of samples in the dataset that are represented by at most one word after OOV cleaning. For the reasons explained above, it is expected that the success in detecting intent may be lower in data sets with high OOV_{weak}. In Table 4.2, two measurements (OOV_{mean} and OOV_{weak}) are shown in bold because they are thought to affect success.

4.2. Comparison of Benchmark Datasets Tests

The suggested BLSTM model is applied to the data prepared using the processes outlined in Chapter 3.2, and the system is tested using 5-fold cross-validation. For comparison purposes, the test results obtained are presented in Table 4.3 together with other literature studies using the same data sets.

Table 4.3. Comparative status of the proposed study in the literature

Reference	Learning Model	Vector Model	Datasets					
			ATIS	**Snips**	**Face book**	**Web App**	**Ask Ubuntu**	**Chat bot**
This Study	LSTM	Generalized SemSpace	**99.78**	98.90	**99.91**	**94.81**	**95.82**	95.12
Shridhar et al., 2019	<Multi>	TF-IDF	--	--	--	**85.00**	**94.00**	**99.60**
Balodis and Deksne, 2019	NN + CNN	FastText	--	--	--	84.75	92.66	98.49
Braun et al., 2017	NLU services		--	--	--	73.00	86.00	94.30
Ren and Xue, 2020	Fusion Features	Word2Vec	99.56	**99.31**	**99.28**	--	-	-
Zhang and et al., 2019	BERT-SLU	WordPiece	**99.76**	98.96	98.88	--		-
Schuster and et al., 2018	Cross-Lingual	Context vectors	-	-	99.11	--		-
Wang et al., 2018	Bi-model based RNN	Context vectors	98.99	-	-	--		-
Chen et al., 2019	Joint BERT+CRF	WordPiece	97.90	98.60	-	--		-
Kim et al., 2016	BLSTM	GloVe	97.31	-	-	--		-
Hakkani et al., 2016	BLSTM	Word2Vec	94.70	-	-	--		-
Zhang et al., 2018	CAPSULE-NLU	Xavier initializer	95.00	97.70	-	--		-
Liu and Lane, 2016	Attention-BiRNN	Context vectors	91.10	96.70	97.30	--		-
Goo and et al., 2018	Slot-Gated Full-Attn	Context vectors	94.10	97.00	95.43	--		-
Firdaus et. al., 2023	Attention + mBERT + CRF		99.18	99.11				

Table 4.3 lists the methods used by the selected studies for comparison and their success on the popular six datasets. In the Table 4.3, bolded values represent the highest performance values attained to date and the highest success values attained in this thesis. The proposed model's performance is measured using the 5-fold cross-validation method. Since 5-fold is used, the model's effectiveness is demonstrated by averaging the results of five different test sets. However, in the validity method based on random sampling used in some studies (Ren and Xue, 2020), success is usually achieved from a single test set. This situation makes it challenging to comment on the

effectiveness of the suggested model across various test sets, especially when we want to evaluate how various test conditions affect performance. According to the data in Table 4.3, the model proposed in this study has achieved the highest success in many datasets in the literature. However, the ChatBot dataset, which has the highest OOV_{weak} and the lowest OOV_{mean} values, has been determined as the dataset in which the proposed model performs the weakest compared to the literature. Detailed analyses shows that the low success in the Chatbot dataset is due to the OOV problem that occurs as a result of the loss of a large number of words during feature extraction. Therefore, it can be concluded that the proposed model may have limitations in achieving the expected performance on datasets that involve more OOV problems. Similarly, it can be observed that the success of the system increases with the increase of the OOV_{mean} value. In the presented study, it can be suggested that to address the OOV issue, the relational data used in the generalized SemSpace training should be augmented with the vocabulary from the intent detection dataset. Expressed from a different perspective, it is possible to state that the 300D generalized SemSpace vectors used in this study are based on approximately 147,000 words of WordNet data, while the 300D GloVe vectors, successfully used in many studies, are derived from a word pool containing 2.2 million words. Therefore, it is obvious that the success of intent detection can be considerably impacted by using a larger vocabulary for generalized SemSpace training.

4.3. Effects of Dataset Size on Classification Time

In deep learning applications, it is an important factor known that the learning process is affected by the size of the dataset. It has been observed that as the dataset size increases, the learning performance tends to increase. However, increasing the size of the dataset can also raise the issue that the computational complexity of the proposed algorithm may increase.

The varying training times depending on the size of each data set used in this study are presented graphically in Figure 4.1.

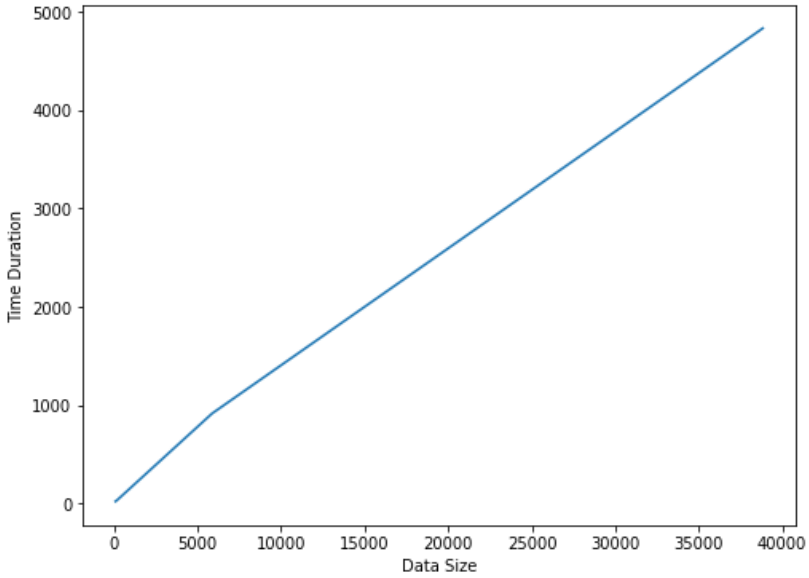

Figure 4.1. The training time periods

It has been observed that the training time does not show an exponential increase depending on the change in data size. This shows that an increase in dataset size will not cause a problem in terms of training time.

5. CONCLUSIONS

In this study, the effects of using vectors based on a new sense-embedding approach in the field of intent detection, called generalized SemSpace, is examined, and a new model called "contextualized Deep SemSpace" is proposed. The proposed model has a three-stage structure: (1) the creation of synset vectors using the generalized SemSpace algorithm, which is a novel algorithm using only WordNet data, (2) a distance-based contextualization approach, (3) a deep learning method such as BLSTM. Six frequently used intent detection data sets are used to evaluate the proposed model, and the results are compared with those from similar studies. The choice of the appropriate synset vector is necessary because the generalized SemSpace technique focuses on senses rather than lemmas, and therefore a WSD approach is used to detect the correct sysnet vectors. In the WSD method used, after a cluster representing the context is identified, the closest senses of the ambiguous words to this cluster are found using Euclidean distance. Even though we worked on closed-domain datasets having only one context cluster, it is seen that a kind of topic clustering process is applied over open-domain datasets with many different topics (Yin et al., 2023). Therefore, it is envisaged that the context cluster approach proposed for closed-domain in this study can also be adapted to open-domain dialogue systems. However, due to the OOV problem, it isn't possible to use every word in the classification step. Vectors obtained with the correct senses increased the success rate after training with the BLSTM model, and almost the best results in the literature were obtained. It has been noted that the suggested system can easily attain the highest classification accuracy in existing literature; however, its performance is notably affected by the OOV issue. Considering that determining intent with just a few words is difficult even for human intelligence, this sensitivity of the method should not be seen as a disadvantage. By reducing OOV related issues, that is, by enriching the data, it is highly likely that the system's success will be further increased.

Thanks to factors such as both using semantic vectors and creating a context-cluster-based WSD, the proposed approach has been able to generate context-dependent word vectors like BERT, ELMo, and GPT. Therefore, it can be concluded that the proposed approach is a very successful intent classifier.

Although the best results obtained in the literature is achieved in the study, two basic problems are encountered and the development of the contextualized Deep SemSpace model will be beneficial for future research. The first problem is related to the processing time of the WSD approach used. Especially during the determination of the context cluster, calculation of combinatorial states takes a significant amount of time. Alternative approaches need to be evaluated in order to shorten this time. On the other hand, the dependency of synset vectors on WordNet in terms of vocabulary makes the OOV problem inevitable. This problem sometimes causes situations where there are no words left due to OOV, especially in data sets where the number of words in the sentence is relatively low. In order to overcome this problem, it is considered that identifying some

words in the existing vocabulary that have semantic relationships with OOV words may be sufficient. In these new relationships, the fact that semantic relationship types defined in WordNet are not specified may be tolerable by the generalized SemSpace method. For example, by using word embedding methods such as Word2Vec, FastText, BERT, ELMo neighboring words of these OOV words can be determined and studies can be carried out using these relational words instead of OOV words that are not considered at all in the calculations. Another suggestion is to generate new word embeddings for these OOV words in the generalized SemSpace space based on their context. The equivalent of each OOV word in the generalized SemSpace can be determined, and then it can be examined whether these detected equivalents form a meaningful cluster. By structuring these clusters appropriately for OOV words, the average of the vectors in these clusters can be calculated, and a new representation value can be assigned to the OOV word.

Since the OOV words in our study are not included in the generalized SemSpace definitions, they cannot be converted into synset vectors and therefore are not taken into account in the calculations. So, it is inevitable that increasing the number of OOV words in sentences may reduce the classification success of the contextualized Deep SemSpace model. Although more advanced models such as large language models (LLMs) have emerged in the literature for the intent detection problem in recent years, the main aim of these models is to try to learn human expertise. However, when we examine WordNet based models, as we use in our model, the rules of human expertise already exist. For OOV problem, only additions will be made to the existing Wordnet. However, in LLMs, systems and expertise are also expensive. Additionally, large corpora are needed to train these models. Therefore, it is clear that future studies will be significant in terms of calculations and costs to continue working on the solution of OOV problem in our approach. Morover, although synset vectors are vocabulary dependent on WordNet, in our study, most datasets achieve the same or better performance than the literature. Therefore, it is possible that solving the OOV problem with additions to WordNet will increase the success of our proposed system system at a lower cost compared to LLMs.

At the same time, since generalized SemSpace vectors are dependent on Wordnet, it causes most words required for slot filling to be evaluated as OOV. Because the task called slot filling is the definition of words corresponding to certain parameters of a user query. In other words, slot values are important pieces of information to be extracted from the user's statement in order to produce accurate responses to it. Since these pieces of information generally contain concepts such as proper names and these concepts are not available in WordNet. It is predicted that applying the generalized SemSpace method with this version to the slot filling problem would not yield successful results. However, for OOV words, it is thought that the generalized SemSpace method will increase the success in the slot filling problem as a result of improvements as mentioned above. In these respects, it is expected that the proposed study will shed light on many future studies.

REFERENCES

Ayanouz, S., Abdelhakim, B. A., & Benhmed, M. (2020, March). A smart chatbot architecture based NLP and machine learning for health care assistance. In *Proceedings of the 3rd international conference on networking, information systems & security* (pp. 1-6).

Banerjee, S., & Pedersen, T. (2002, February). An adapted Lesk algorithm for word sense disambiguation using WordNet. In *International conference on intelligent text processing and computational linguistics* (pp. 136-145). Berlin, Heidelberg: Springer Berlin Heidelberg.

Basile, P., Caputo, A., & Semeraro, G. (2014, August). An enhanced lesk word sense disambiguation algorithm through a distributional semantic model. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers* (pp. 1591-1600).

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks, 5*(2), 157-166.

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the association for computational linguistics, 5*, 135-146.

Brody, S., & Lapata, M. (2009, March). Bayesian word sense induction. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)* (pp. 103-111).

Camacho-Collados, J., & Pilehvar, M. T. (2018). From word to sense embeddings: A survey on vector representations of meaning. *Journal of Artificial Intelligence Research, 63*, 743-788.

Camacho-Collados, J., Pilehvar, M. T., & Navigli, R. (2016). Nasari: Integrating explicit knowledge and corpus statistics for a multilingual representation of concepts and entities. *Artificial Intelligence, 240*, 36-64.

Cao, Y., Liu, F., Simpson, P., Antieau, L., Bennett, A., Cimino, J. J., ... & Yu, H. (2011). AskHERMES: An online question answering system for complex clinical questions. *Journal of biomedical informatics, 44*(2), 277-288.

Celikyilmaz, A., Sarikaya, R., Hakkani-Tür, D., Liu, X., Ramesh, N., & Tür, G. (2016, September). A New Pre-Training Method for Training Deep Learning Models with Application to Spoken Language Understanding. In *Interspeech* (pp. 3255-3259).

Chandra, G., & Dwivedi, S. K. (2014, December). A literature survey on various approaches of word sense disambiguation. In *2014 2nd International Symposium on Computational and Business Intelligence* (pp. 106-109). IEEE.

Chen, D., Huang, Z., Wu, X., Ge, S., & Zou, Y. (2022). Towards joint intent detection and slot filling via higher-order attention. In *IJCAI*.

Chen, X., Liu, Z., & Sun, M. (2014, October). A unified model for word sense representation and disambiguation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1025-1035).

Chen, X., Xu, L., Liu, Z., Sun, M., & Luan, H. (2015, June). Joint learning of character and word embeddings. In *Twenty-fourth international joint conference on artificial intelligence*.

Chen, Y. N., Wang, W. Y., & Rudnicky, A. I. (2013, December). Unsupervised induction and filling of semantic slots for spoken dialogue systems using frame-semantic parsing. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding* (pp. 120-125). IEEE.

Chowdhary, K., & Chowdhary, K. R. (2020). Natural language processing. *Fundamentals of artificial intelligence*, 603-649.

Coucke, A., Saade, A., Ball, A., Bluche, T., Caulier, A., Leroy, D., ... & Dureau, J. (2018). Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*.

Cuayáhuitl, H. (2017). Simpleds: A simple deep reinforcement learning dialogue system. *Dialogues with Social Robots: Enablements, Analyses, and Evaluation*, 109-118.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dowding, J., Gawron, J. M., Appelt, D., Bear, J., Cherny, L., Moore, R., & Moran, D. (1994). Gemini: A natural language system for spoken-language understanding. *arXiv preprint cmp-lg/9407007*.

Edmonds, P., & Agirre, E. E. (2006). Word Sense Disambiguation: Algorithms And Applications.

El-Alami, F. Z., El Alaoui, S. O., & Nahnahi, N. E. (2022). Contextual semantic embeddings based on fine-tuned AraBERT model for Arabic text multi-class categorization. Journal of King Saud University-Computer and Information Sciences, 34(10), 8422-8428.

Etzioni, O., Cafarella, M., Downey, D., Kok, S., Popescu, A. M., Shaked, T., ... & Yates, A. (2004, May). Web-scale information extraction in knowitall: (preliminary results). In *Proceedings of the 13th international conference on World Wide Web* (pp. 100-110).

Fellbaum, C. (1998). A semantic network of english: the mother of all WordNets. *Computers and the Humanities*, *32*, 209-220.

Firdaus, M., Ekbal, A., & Cambria, E. (2023). Multitask learning for multilingual intent detection and slot filling in dialogue systems. *Information Fusion*, *91*, 299-315.

Firdaus, M., Golchha, H., Ekbal, A., & Bhattacharyya, P. (2021). A deep multi-task model for dialogue act classification, intent detection and slot filling. *Cognitive Computation*, *13*, 626-645.

Gers, F. A., Schraudolph, N. N., & Schmidhuber, J. (2002). Learning precise timing with LSTM recurrent networks. *Journal of machine learning research*, *3*(Aug), 115-143.

Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural networks*, *18*(5-6), 602-610.

Hakkani-Tür, D., Tür, G., Celikyilmaz, A., Chen, Y. N., Gao, J., Deng, L., & Wang, Y. Y. (2016, September). Multi-domain joint semantic frame parsing using bi-directional rnn-lstm. In *Interspeech* (pp. 715-719).

Han, S. C., Long, S., Li, H., Weld, H., & Poon, J. (2022). Bi-directional joint neural networks for intent classification and slot filling. *arXiv preprint arXiv:2202.13079*.

Heck, L., & Hakkani-Tür, D. (2012, December). Exploiting the semantic web for unsupervised spoken language understanding. In *2012 IEEE Spoken Language Technology Workshop (SLT)* (pp. 228-233). IEEE.

Hemphill, C. T., Godfrey, J. J., & Doddington, G. R. (1990). The ATIS spoken language systems pilot corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990*.

Hirschberg, J., & Manning, C. D. (2015). Advances in natural language processing. *Science*, *349*(6245), 261-266.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735-1780.

Hui, Y., Wang, J., Cheng, N., Yu, F., Wu, T., & Xiao, J. (2021, June). Joint intent detection and slot filling based on continual learning model. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 7643-7647). IEEE.

Iacobacci, I. J., Pilehvar, M. T., & Navigli, R. (2016). Embeddings for word sense disambiguation: An evaluation study. In *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016-Long Papers* (Vol. 2, pp. 897-907). Association for Computational Linguistics (ACL).

Jokinen, K., & McTear, M. (2022). Spoken dialogue systems. Springer Nature.

Jauhar, S. K., Dyer, C., & Hovy, E. (2015). Ontologically grounded multi-sense representation learning for semantic vector space models. In *proceedings of the 2015 conference of the north American chapter of the Association for Computational Linguistics: human language technologies* (pp. 683-693).

Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2016). Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Kawahara, T. (2019). Spoken dialogue system for a human-like conversational robot ERICA. In *9th International Workshop on Spoken Dialogue System Technology* (pp. 65-75). Springer Singapore.

Kim, J. K., Tur, G., Celikyilmaz, A., Cao, B., & Wang, Y. Y. (2016, December). Intent detection using semantically enriched word embeddings. In *2016 IEEE spoken language technology workshop (SLT)* (pp. 414-419). IEEE.

Korpusik, M., & Glass, J. (2017). Spoken language understanding for a nutrition dialogue system. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *25*(7), 1450-1461.

Lai, S., Xu, L., Liu, K., & Zhao, J. (2015, February). Recurrent convolutional neural networks for text classification. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 29, No. 1).

Lesk, M. (1986, June). 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* (pp. 24-26).

Li, J., & Jurafsky, D. (2015). Do multi-sense embeddings improve natural language understanding?. *arXiv preprint arXiv:1506.01070*.

Lilleberg, J., Zhu, Y., & Zhang, Y. (2015, July). Support vector machines and word2vec for text classification with semantic features. In *2015 IEEE 14th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC)* (pp. 136-140). IEEE.

Lochter, J. V., Silva, R. M., & Almeida, T. A. (2020, October). Deep learning models for representing out-of-vocabulary words. In *Brazilian Conference on Intelligent Systems* (pp. 418-434). Cham: Springer International Publishing.

Lu, Y., Zhao, Y., Cui, R., & Jin, G. (2023, April). Research on the Joint Learning Method of Intent Detection and Slot Filling by Fusing Tag Semantic Information. In *2023 IEEE International Conference on Control, Electronics and Computer Technology (ICCECT)* (pp. 838-842). IEEE.

Ma, Z., Sun, B., & Li, S. (2022). A Two-Stage Selective Fusion Framework for Joint Intent Detection and Slot Filling. *IEEE Transactions on Neural Networks and Learning Systems*.

Manning, C., & Schutze, H. (1999). *Foundations of statistical natural language processing*. MIT press.

Mesnil, G., Dauphin, Y., Yao, K., Bengio, Y., Deng, L., Hakkani-Tur, D., ... & Zweig, G. (2014). Using recurrent neural networks for slot filling in spoken language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *23*(3), 530-539.

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

Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., & Miller, K. J. (1990). Introduction to WordNet: An on-line lexical database. *International journal of lexicography*, *3*(4), 235-244.

Nadkarni, P. M., Ohno-Machado, L., & Chapman, W. W. (2011). Natural language processing: an introduction. *Journal of the American Medical Informatics Association*, *18*(5), 544-551.

Navigli, R. (2009). Word sense disambiguation: A survey. *ACM computing surveys (CSUR)*, *41*(2), 1-69.

Navigli, R., & Velardi, P. (2005). Structural semantic interconnections: a knowledge-based approach to word sense disambiguation. *IEEE transactions on pattern analysis and machine intelligence*, *27*(7), 1075-1086.

Ng, H. T., & Lee, H. B. (1996). Integrating multiple knowledge sources to disambiguate word sense: An exemplar-based approach. *arXiv preprint cmp-lg/9606032*.

Ni, J., Young, T., Pandelea, V., Xue, F., & Cambria, E. (2023). Recent advances in deep learning based dialogue systems: A systematic survey. *Artificial intelligence review*, *56*(4), 3055-3155.

Niu, C., Li, W., Srihari, R. K., Li, H., & Crist, L. (2004, July). Context clustering for word sense disambiguation based on modeling pairwise context similarities. In *Proceedings of SENSEVAL-3, the third international workshop on the evaluation of systems for the semantic analysis of text* (pp. 187-190).

Orhan, U., & Tulu, C. N. (2021). A novel embedding approach to learn word vectors by weighting semantic relations: SemSpace. *Expert Systems with Applications*, *180*, 115146.

Pal, A. R., & Saha, D. (2015). Word sense disambiguation: A survey. *arXiv preprint arXiv:1508.01346*.

Pantel, P., & Lin, D. (2002, July). Discovering word senses from text. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 613-619).

Pennington, J., Socher, R., & Manning, C. D. (2014, October). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532-1543).

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, *1*(8), 9.

Raymond, C., & Riccardi, G. (2007, August). Generative and discriminative algorithms for spoken language understanding. In *Interspeech 2007-8th Annual Conference of the International Speech Communication Association*.

Ren, F., & Xue, S. (2020). Intention detection based on siamese neural network with triplet loss. *IEEE Access*, *8*, 82242-82254.

Rothe, S., & Schütze, H. (2015). Autoextend: Extending word embeddings to embeddings for synsets and lexemes. *arXiv preprint arXiv:1507.01127*.

Samin, A. M., Kobir, M. H., Rafee, M. M. S., Ahmed, M. F., Kibria, S., & Rahman, M. S. (2022). Investigating the effect of domain selection on automatic speech recognition performance: a case study on Bangladeshi Bangla. arXiv preprint arXiv:2210.12921.

Sarikaya, R., Crook, P. A., Marin, A., Jeong, M., Robichaud, J. P., Celikyilmaz, A., ... & Radostev, V. (2016, December). An overview of end-to-end language understanding and dialog management for personal digital assistants. In *2016 ieee spoken language technology workshop (slt)* (pp. 391-397). IEEE.

Sarzynska-Wawer, J., Wawer, A., Pawlak, A., Szymanowska, J., Stefaniak, I., Jarkiewicz, M., & Okruszek, L. (2021). Detecting formal thought disorder by deep contextualized word representations. *Psychiatry Research*, *304*, 114135.

Schuster, S., Gupta, S., Shah, R., & Lewis, M. (2018). Cross-lingual transfer learning for multilingual task oriented dialog. *arXiv preprint arXiv:1810.13327*.

Seker, A., Bandel, E., Bareket, D., Brusilovsky, I., Greenfeld, R., & Tsarfaty, R. (2022, May). AlephBERT: Language model pre-training and evaluation from sub-word to sentence level. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 46-56).

Seo, J., Lee, S., Liu, L., & Choi, W. (2022). TA-SBERT: Token attention sentence-BERT for improving sentence representation. IEEE Access, 10, 39119-39128.

Serban, I., Sordoni, A., Bengio, Y., Courville, A., & Pineau, J. (2016, March). Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 30, No. 1).

Seyyar, Y. E., Yavuz, A. G., & Ünver, H. M. (2022). An attack detection framework based on BERT and deep learning. IEEE Access, 10, 68633-68644.

Sinoara, R. A., Camacho-Collados, J., Rossi, R. G., Navigli, R., & Rezende, S. O. (2019). Knowledge-enhanced document embeddings for text classification. *Knowledge-Based Systems*, *163*, 955-971.

Song, L. (2016). Word embeddings, sense embeddings and their application to word sense induction. *The University of Rochester, April*.

Sun, C., Lv, L., Liu, T., & Li, T. (2022). A joint model based on interactive gate mechanism for spoken language understanding. *Applied Intelligence*, *52*(6), 6057-6064.

Tong, Y., Guo, J., & Zhou, J. (2022). Separation inference: A unified framework for word segmentation in east asian languages. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 30, 1521-1530.

Tu, N. A., Hieu, D. X., Phuong, T. M., & Bach, N. X. (2023, June). A Bidirectional Joint Model for Spoken Language Understanding. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 1-5). IEEE.

Tulu, C. (2019). Semantic vector space model using Euclidean distance based relatedness: SemSpace, Ph.D. dissertation, Department of Computer Enginering, Cukurova Universty., Adana, Turkey.

Tur, G., Hakkani-Tür, D., & Heck, L. (2010, December). What is left to be understood in ATIS?. In *2010 IEEE Spoken Language Technology Workshop* (pp. 19-24). IEEE.

Tur, G., Hakkani-Tür, D., & Schapire, R. E. (2005). Combining active and semi-supervised learning for spoken language understanding. *Speech Communication*, *45*(2), 171-186.

Tur, G., Hakkani-Tür, D., Heck, L., & Parthasarathy, S. (2011, May). Sentence simplification for spoken language understanding. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 5628-5631). IEEE.

Wang, S., Huang, M., & Deng, Z. (2018, July). Densely connected CNN with multi-scale feature attention for text classification. In *IJCAI* (Vol. 18, pp. 4468-4474).

Wei, P., Zeng, B., & Liao, W. (2022). Joint intent detection and slot filling with wheel-graph attention networks. *Journal of Intelligent & Fuzzy Systems*, *42*(3), 2409-2420.

Weld, H., Huang, X., Long, S., Poon, J., & Han, S. C. A survey of joint intent detection and slot-filling models in natural language understanding. arXiv 2021. *arXiv preprint arXiv:2101.08091*.

Xu, P., & Sarikaya, R. (2013, December). Convolutional neural network based triangular crf for joint intent detection and slot filling. In *2013 ieee workshop on automatic speech recognition and understanding* (pp. 78-83). IEEE.

Yao, L., Mao, C., & Luo, Y. (2019, July). Graph convolutional networks for text classification. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 33, No. 01, pp. 7370-7377).

Yao, K., Peng, B., Zhang, Y., Yu, D., Zweig, G., & Shi, Y. (2014, December). Spoken language understanding using long short-term memory neural networks. In *2014 IEEE Spoken Language Technology Workshop (SLT)* (pp. 189-194). IEEE.

Yao, K., Peng, B., Zweig, G., Yu, D., Li, X., & Gao, F. (2014, May). Recurrent conditional random field for language understanding. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 4077-4081). IEEE.

Yarowsky, D. (1995, June). Unsupervised word sense disambiguation rivaling supervised methods. In *33rd annual meeting of the association for computational linguistics* (pp. 189-196).

Yin, C., Li, P., & Ren, Z. (2023, April). CTRLStruct: Dialogue Structure Learning for Open-Domain Response Generation. In Proceedings of the ACM Web Conference 2023 (pp. 1539-1550).

Zhang, M., Mosbach, M., Adelani, D. I., Hedderich, M. A., & Klakow, D. (2022). Mcse: Multimodal contrastive learning of sentence embeddings. arXiv preprint arXiv:2204.10931.

Zhang, X., Cai, F., Hu, X., Zheng, J., & Chen, H. (2022). A Contrastive learning-based Task Adaptation model for few-shot intent recognition. *Information Processing & Management*, *59*(3), 102863.

Zhang, X., & Wang, H. (2016, July). A joint model of intent determination and slot filling for spoken language understanding. In *IJCAI* (Vol. 16, No. 2016, pp. 2993-2999).

Zhang, S., Wei, Z., Wang, Y., & Liao, T. (2018). Sentiment analysis of Chinese micro-blog text based on extended sentiment dictionary. *Future Generation Computer Systems*, *81*, 395-403.

Zhao, J., Yin, S., & Xu, W. (2022, January). Attention-based iterated dilated convolutional neural networks for joint intent classification and slot filling. In *CCF Conference on Big Data* (pp. 150-162). Singapore: Springer Singapore.

Zhang, X., Zhao, J., & LeCun, Y. (2015). Character-level convolutional networks for text classification. *Advances in neural information processing systems*, *28*.

CURRICULUM VITAE

Elif Gülfidan TOSUN, received a Bachelor's degree in 2014 and M.Sc. degree in 2017 from Pamukkale University Department of Computer Engineering. After graduation, in 2014, she started to work at Mersin University as a research assistant. Now, she has been working as Lecturer in Pamukkale University since 2019.