Improving Intent Classification By Automatic Data Augmentation Using Word Sense Disambiguation

by

Prashant Garg

A Thesis Presented in Partial Fulfillment
of the Requirement for the Degree
Master of Science

Approved November 2018 by the
Graduate Supervisory Committee:

Chitta Baral, Chair
Hemanth Kumar
Yezhou Yang

ARIZONA STATE UNIVERSITY

December 2018

ABSTRACT

Virtual digital assistants are automated software systems which assist in understanding natural languages such as English, either in voice or textual form. In recent times, a lot of digital applications have shifted towards providing a user experience using natural language interface. The change is brought up by the degree of ease with which the virtual digital assistants such as Google Assistant and Amazon Alexa can be integrated into your application. These assistants make use of a Natural Language Understanding (NLU) system which acts as an interface to translate unstructured natural language data into a structured form. Such an NLU system uses an intent finding algorithm which gives a high-level idea or meaning of a user query, termed as intent classification. The intent classification step identifies the action(s) that a user wants the assistant to perform. The intent classification step is followed by an entity recognition step in which the entities in the utterance are identified on which the intended action is performed. This step can be viewed as a sequence labeling task which maps an input word sequence into a corresponding sequence of slot labels. This step is also termed as slot filling.

In this thesis, we improve the intent classification and slot filling in the virtual voice agents by automatic data augmentation. Spoken Language Understanding systems face the issue of data sparsity. The reason behind this is that it is hard for a human created training sample to represent all the patterns in the language. Due to the lack of relevant data, deep learning methods are unable to generalize the Spoken Language Understanding model. This thesis expounds a way to overcome the issue of data sparsity in deep learning approaches on Spoken Language Understanding tasks. Here we have described the limitations in the current intent classifiers and how the proposed algorithm uses existing knowledge bases to overcome those limitations. The method helps in creating a more robust intent classifier and slot filling system.

To Mom and Dad

ACKNOWLEDGMENTS

Here, I would like to bestow my gratitude to the people involved in this project for their continuous and indispensable support and guidance. First and foremost, my thesis advisor, Dr. Chitta Baral, who has guided me through various milestones of this project with his expertise and knowledge. Also, my committee members Yezhou Yang and Hemanth Kumar, who have offered their guidance and support as and when needed. Thanks to Arizona State University for providing an opportunity to work with ingenious minds and on cutting-edge technology. Special thanks to my colleagues, especially Arpit Sharma (a Ph.D. scholar) under Dr. Chitta Baral. He kept me motivated through the course of the project and gave valuable inputs. Last but not least, thanks to my parents, and numerous friends who have endured this long journey with me and eased it with their constant support and love.

TABLE OF CONTENTS

Page

CHAPTER Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Chapter 1

INTRODUCTION AND MOTIVATION

1.1 Introduction

Virtual digital assistants are automated software systems which assist in understanding natural languages such as English, either in voice or textual form. Many smart digital devices that are used these days contain these virtual digital assistants. Such digital assistants can be easily seen on our smartphones, namely Siri [1] (Apple), Google Assistant [2] (Google) or smart speakers like Alexa [3] (Amazon) in Echo Speakers.

A market study of growth on virtual assistants reports that the number of active users who are willing to use these assistants on a daily basis are predicted to be 1.6 billion by the end of 2020. The market of the virtual assistant is predicted to reach an all-time high of $12 billion US dollars by 2020 [4] . This data shows that in the coming future, the scope of this project can be of immense importance. The reason behind wide acceptance of these virtual agents is their implementation of natural language interface. Communicating in a natural language such as English, with a machine is more intuitive than interacting using web or mobile interfaces. The learning curve in natural language is minimal as compared to any web interface or mobile, where you have variety of icons to remember, whose representation is susceptible to change in newer versions. So for a common user, the interaction with a machine using natural

[1] `https://www.apple.com/siri`

[2] `https://assistant.google.com`

[3] `https://alexa.amazon.com`

[4] $https://venturebeat.com/2017/10/31/why-digital-assistants-are-so-hot-right-now$

language is much simpler.

Lately, these assistants have started to understand more than one language. Popular assistants such as Amazon Alexa [5] and Google Assistant [6] can now understand English, German and Japanese etc. Not very long ago, under project Google Duplex, Google has expanded to support more than 30 new languages and has presented a successful demo for its multilingual support [7] . All these factors point towards a brighter future for virtual assistants. These assistants are available on a wide range of devices.

Following are the advantages of using virtual voice assistants:

1. **Personalizing**: They are driven by a complex AI engine which keeps track of customer behavior. As a result, when you search through the service, the recommendation system in virtual assistants filters and ranks results based on users past preferences.

2. **Rich knowledge base**: The knowledge base of virtual assistants covers a vast spectrum of data. They can provide knowledge of generic search queries such as *"what is the weather today"* as well as field-specific data such as *"reorder my coffee from Starbucks"* using data stored in the integrated databases of the third-party applications.

3. **One click integration using cloud**:These virtual assistants now have their own application store, where you only need to start a service once, such as support for calendar services or taxi services, and it integrates with the assistant

[5] `https://alexa.amazon.com`

[6] `https://assistant.google.com`

[7] `https://money.cnn.com/2018/08/30/technology/google-assistant-bilingual/index.html`

without requiring any memory on your device. Currently, Amazon Alexa [8] has more than 30,000 skills in its application store.

4. **New levels of connectivity**: The integration of the assistant with different hardware devices and their unique connectivity with various internet-of-things (IoT) devices has created a whole new level of communication between devices, people and businesses.

   According to BI Intelligence report, more than 24 billion intelligent IoT devices will be installed by 2020. These devices will span through the areas of transport, connected homes, various delivery services, etc. Not only can one order a service with your virtual assistant, but you can also track the status of various services.

5. **Enhanced productivity**: Most of these assistants have auto-fill capabilities, which can process your conversation on various platforms and can auto-fill various forms on your behalf. For example, today some company-specific assistants can read a conversation happening with the client and automate the process of scheduling a meeting.

Technology is rapidly changing and these assistants help people in adapting to new technology by providing an easy to use natural language interface.

The working of the virtual digital assistant can be divided into two components, Natural Language Understanding (NLU) system and Dialog Management system (DM). The flow of the conversation can be seen in the figure 1.1 on the page 4. A Natural Language Understanding system is responsible for converting unstructured data into a structured form. A typical NLU system, given a user utterance in natural language in either a transcript of voice or a direct text, detects user intent behind the utterance. An NLU system uses an intent finding algorithm which gives high-level

[8] `https://alexa.amazon.com`

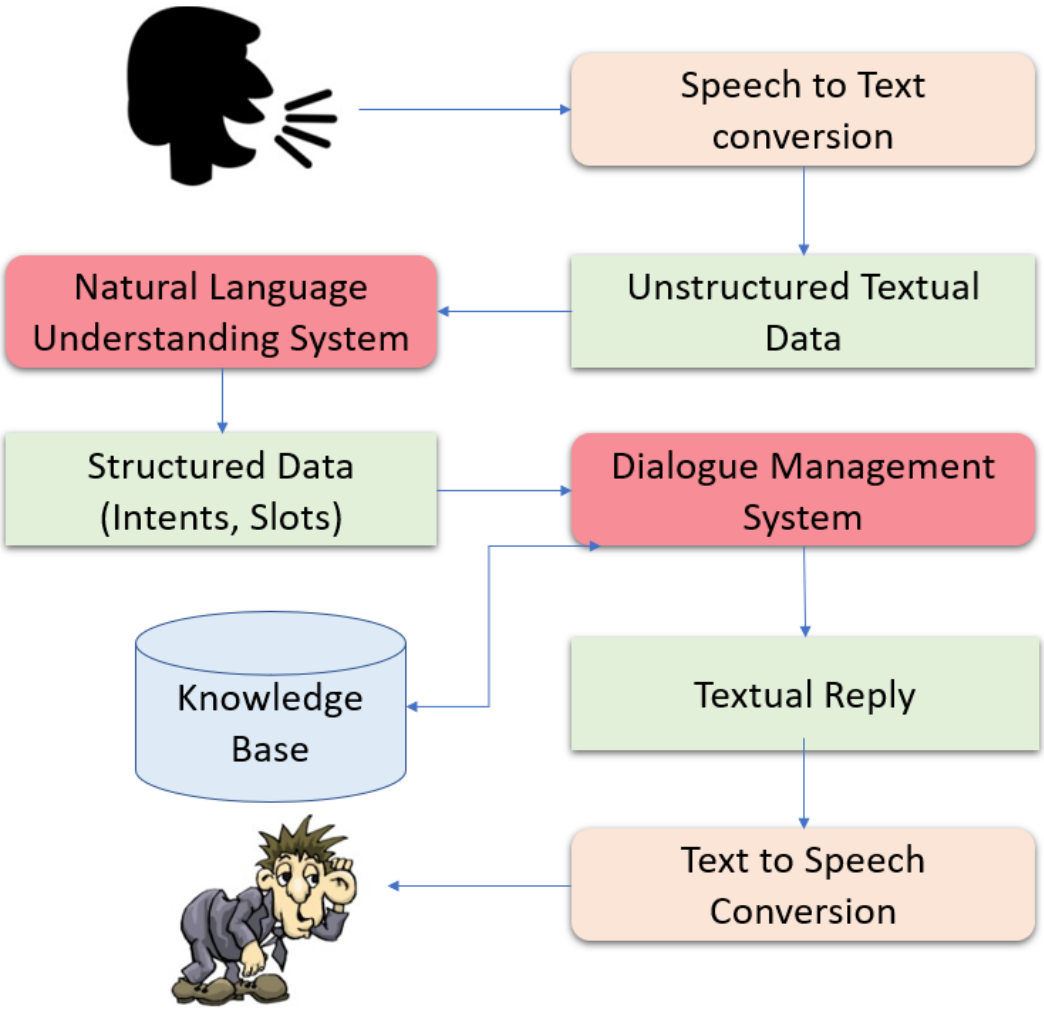


Figure 1.1: Virtual Digital Assistant Overview

idea or meaning of a user query, termed as intent classification.

For example:

1. User Utterance: *"I want to book a flight from Phoenix to Boston."*

   Intent: "Flight Booking".

2. User Utterance: *"Find comedies by Jim Carrey"*

   Intent: "Find Movie".

The second step, followed by intent classification is slot filling, this step may or

may not be done after intent classification, depending upon different algorithms. But one is followed by the other. Slot filling mechanism can be understood as finding a mapping in a user utterance, such that given a user utterance X (x_1 x_2 x_3 x_4 ..... x_T) , if a word sequence from m to n in X (where m >= 1, $n <= T$, $n >= m$) corresponds to a predefined slot label, then we replace the word sequence with a more generalized form. For example:

1. User Utterance: *"I want to book a flight from Phoenix to Boston."*
   Slot: "CITY-1: 'Phoenix' " and "CITY-2: 'Boston' ".

2. User Utterance: *"Find comedies by Jim Carrey"*
   Slot: "Genre: 'Comedy' " and "Artist: 'Jim Carrey'"

The above slot filling mechanism and intent classification together, along with some metadata, forms the NLU System. It is the machine translation part, where you translate your unstructured data into a structured form. In other words, the NLU system acts as a parser to parse unstructured natural language input into a structured form.
For example:

1. User Utterance: *"I want to book a flight from Phoenix to Boston."*

```
A typical output from NLU system will be:
{
        Intent : Flight_Booking,
        Slot :
        {
                CITY-1 : Phoenix,
                CITY-2 : Boston
```

```
        },
        Metadata :
        {
                confidence: 1.0,
                ...................
                ...................
        }
}
```

2. User Utterance: *"Find comedies by Jim Carrey"*

```
A typical output from NLU system will be:
{
        Intent   : Find_Movie,
        Slot:
        {
                Genre : Comedy,
                Artist : Jim Carrey
        },
        Metadata :
        {
                confidence: 1.0,
                ...................
                ...................
        }
}
```

The above representation forms a semantic frame to capture the meaning of user utterance and convert it into a structured form. Some of the famous industrial tools for forming this semantic frame are DialogFlow (Api.ai) [9] , Alexa Skill Set [10] , Luis [11] etc. The ability of this structure to interact with traditional machine services without requiring any major changes makes the intent classification schema a widely accepted and used method. This semantic form can also be seen as a set of actions (Intents) and a set of parameters (Slots), so this form is compatible with all the translation methods which use Action-Parameter framework to convert an unstructured or semi-structured data into machine commands. Some virtual assistants also provide additional features besides intent and slots, such as Fallback (where if you do not meet certain threshold in terms of similarity then you classify them into a category called Fallback), context vector (a tool to represent change in state by the NLU system. Some of the natural language query classification is dependent upon the previous n statements), default value fulfillment (a tool used to define default value of slots if explicit value is not provided in user query), etc.

1.2 Related Work

Until 2011, The Spoken Language Understanding systems considered intent classification and slot filling as two separate problems and used separate techniques for solving them. A good performance was observed in slot filling task by using conditional random fields (CRF). Raymond and Riccardi (2007) used a CRF based discriminative model to perform concepts extraction and segmentation, also know as slot filling mechanism in the context of Spoken language understanding. For intent

[9] `https://dialogflow.com`

[10] `https://developer.amazon.com/alexa`

[11] `https://www.luis.ai/home`

classification, various regression and SVM models (Haffner *et al.* (2003)) were used, in which they optimize support vector machine using a combination of heterogeneous binary classifiers.

In 2013, a joint intent and slot filling model was proposed by Xu and Sarikaya (2013). They introduced a new joint model for intent detection and slot filling based on convolution neural networks. This model can be seen as a triangular conditional random fields model. It was one of the earlier models to use CNN in this domain.

In late 2013, a Recurrent Neural Network (RNN) based architecture emerged for solving NLP problems because of its ability to capture sequence to sequence modeling. Sutskever *et al.* (2014) used multi-layered Long-Short Term Memory (LSTM) model to map input sequence into a vector of a fixed dimensionality, and then another deep LSTM to decode to a vector of a fixed dimensionality. This is also known as encode-decoder Network.

In 2014, Guo *et al.* (2014) used Recursive Neural Networks (RecNNs) to solve intent classification and slot filling task by providing an elegant mechanism for incorporating both discrete syntactic structure and continuous-space word and phrase representations into a powerful compositional model. Since 2014, various RNN architectures are proposed to solve the problem, including Hybrid models between RNNs and CRFs were explored. Mesnil *et al.* (2015) presented a comparison of various RNN techniques with CRF and showed that RNN-based models outperform CRF on the ATIS benchmark.

Peng and Yao (2015) uses RNN with external memory to overcome the issue of gradient vanishing and exploding problem. The paper proposes to use an external memory to improve memorization capability of RNNs.

Kurata *et al.* (2016) paper proposed a modified version of LSTM in which encoder-labeler LSTM is used which first encodes the whole input sequence into a fixed length

vector using encoder LSTM, then uses an encoded vector as an initial state of another LSTM for sequence labeling. Hakkani-Tür *et al.* (2016) proposed an RNN-LSTM architecture for joint modeling of slot filling, intent determination, and domain classification. They have built a joint multi-domain model enabling multi-task deep learning where the data from each domain reinforces each other. Liu and Lane (2016) was one of the first works to use attention based encoder-decoder technique for joint intent detection and slot filling which outperformed every previous model.

Recently, Goo *et al.* (2018) added a slotted gate for encapsulating intent context while learning attention parameters for slot mechanism. This paper has obtained state of the art accuracy on intent classification and slot filling. It is considered as a baseline for this research. This paper has beaten previous state of the art system. In this paper, they added a slotted gate for encapsulating intent context while learning attention parameters for slot mechanism.

1.3 Limitations and Problems in Current Systems

The translation of a natural language utterance into its structured semantic representation is a hard task. The statistical approaches which use word matching algorithms as a subroutine, between user utterance and training samples, fail to represent the semantics of a natural language utterance because with change in only one word in a sentence of n-words, the meaning of the sentence may change entirely and most algorithms use number of words matched as a criteria to classify intent.
For example:

1. *"Hi Assistant, can you help me in selling apples"*.

2. *"Hi Assistant, can you help me in buying apples"*.

The semantic meaning of sentence 1 and sentence 2 is completely different because

of a change in only one word "selling" in sentence 1 to "buying" in sentence 2. This change in few words can change semantic for the NLU system in two ways, either changing the intent of the user utterance or changing the fulfillment of a slot value. The above example shows how a single word can change the intent from "Selling Item" to "Buying Item".

1. *"Hey Assistant, Can you show me comedy movies of Jim Carrey"*
2. *"Hey Assistant, Can you show me comedy movies but of Jim Carrey"*

In both of the above examples, the intent behind both the sentences is "Find Movie", But due to an addition of "but" word in sentence 2 the expected output from user utterance in sentence 1 is different from expected output from sentence 2.

The statistical methods or pattern matching methods used for intent classification fail to encapsulate the semantics of these scenarios. Even today, if you search a query with a negation filter in natural language on famous applications like YouTube, it fails to recognize the semantic difference in two utterances.

The semantic encapsulation in structured form can be categorized into two categories based on the type of error.

1. **Same meaning sentences**: A learning system takes training examples for a domain and learns a target function for classification (Intent, in our case). From this learning system, we expect that if enough training examples of a certain intent class are provided, then the system should be able to predict the right intent of new utterance with high probability. If the new utterance is same in meaning with the training examples of that class, then the learning algorithm should be able to predict the intent irrespective of the words used in a new utterance. The term "enough examples" means a number near to the amount

of training examples required by a human being who understands that language. In the current industrial systems, such as DialogFlow [12] , Luis [13] and Watson [14] , we observe that the probability of prediction of right intent drops significantly when we use same meaning sentences but not same words in a new utterance classification. The details of this experiment can be seen in Appendix A. This shows that if we can incorporate some prior knowledge of the language during training, then we might be able to tackle this issue.

2. **Dissimilar meaning sentences**: In natural language systems, the input from the user can be outside of the domain of an application. In traditional systems, the types of inputs possible have a limited domain. But in the natural language processing system, we do not have a limited domain. So, it becomes important to detect whether a user utterance belongs to the domain of an application or not. This is often referred to as "fallback intent" in NLU systems, which means that the system is not able to classify the given user utterance with a minimum threshold confidence. It is important for classification of a user utterance which does not share same semantics with any of the given class in the application (may or may not share same words), the learning algorithm should be able to detect that it is out of the domain of that application. In the current industrial systems, such as Dialogflow [12], Luis [13] and Watson [14], we observe that the probability of intent prediction on dissimilar sentences, tends towards classes which share same words, in-spite of different semantics. The details of this experiment can be seen in Appendix A.

The above two approaches show that if we can incorporate prior knowledge of the

[12] `https://dialogflow.com`

[13] `https://www.luis.ai`

[14] `https://www.ibm.com/watson/ai-assistant`

language then we can improve the accuracy of the system.

Lack of Relevant Data: Various natural language utterances can be used to convey a common idea. Consequently, it is hard to provide a learning algorithm with enough training examples which represent all the possible scenarios so that it can learn a generalized model of a domain. Whereas humans are able to generalize with only a limited number of examples. One possible reason for this may be that humans use their prior knowledge of the language to generalize faster.

1.4 An Overview to the Approach

The approach is based on using prior knowledge to overcome the limitations of the current Spoken Language Understanding systems. The methodology includes two basic component modules, namely, data augmentation and learning on augmented data. As discussed earlier, due to the high cost of manual human effort required to create annotated data, the spoken language understanding systems face issues of data sparsity. The training examples are not able to capture all the scenarios of the domain. Therefore, deep learning models are not able to generalize well. In our approach, we are using prior language knowledge to artificially extend training data with the more relevant samples. We are performing tokenization on the data and then using part of speech tagging and word sense disambiguation techniques to find new similar and dissimilar sentences of a given training sample. Then, we are dividing them into two parts "similar sentences" and "dissimilar sentences". As the name suggests, similarity model is to encapsulate semantics of same meaning sentences and dissimilarity model is for encapsulating the semantics of dissimilar meaning sentences so that model can make a more informed decision if a user utterance falls out of the domain. If the meaning of the generated sentence is same as of sentence in training data, irrespective of the words used in both sentences, then the sentence is added in "similar sentences"

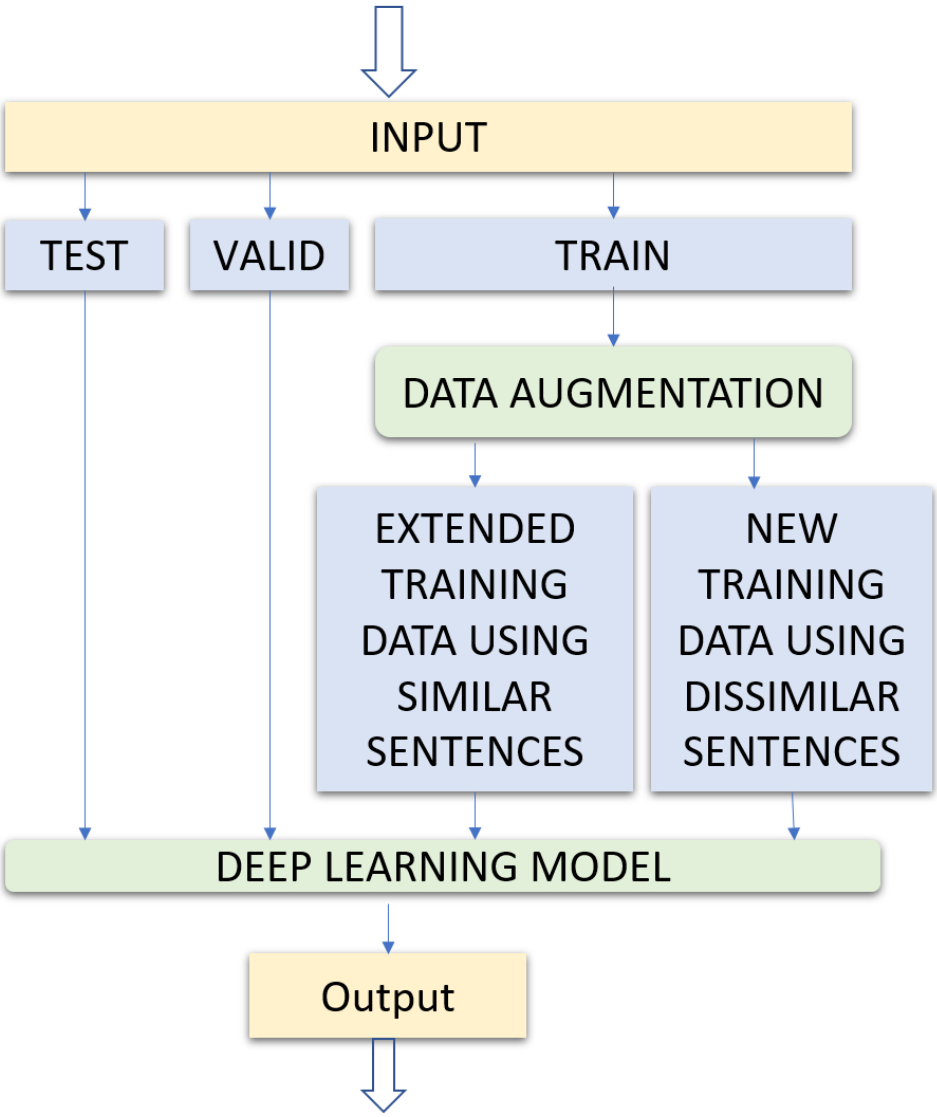


Figure 1.2: Overview of the Proposed System

category and if the generated sentences lie outside the domain of the application then we put that sentence in "dissimilar sentence" category. After generating this artificial annotated data, we used attention based deep learning approach. We run two models in parallel, one on similar sentences termed as "Similarity Model" and other on dissimilar sentences termed as "Dissimilarity Model". Then we pass the output from the models to create a final decider function to classify. This function will return a final structured format of initial utterance.

Chapter 2

DATA AUGMENTATION

2.1 Initial Observation and Motivation

In natural language, there are various ways of saying the same thing. To make a machine learn the language it is important that we need to cover as much as pattern possible, so that machine can generalize better. It's hard for a human created data set to cover up all the scenarios in the training set. In general, given few instances a human who has a decent knowledge of the language is able to generalize faster. The reason is humans use prior knowledge of the language to generalize faster.

Some deep learning models are data hungry models. The models used in SLU task such as Recurrent Neural Network (RNN) are also data hungry as they try to generalize on training samples. The problem is that manually annotated training samples in SLU systems may not capture all the scenarios needed to generalize the model.

Dataset Name	ATIS	SNIPS
Vocabulary Size	722	11,241
Number of slots	120	72
Number of Intents	21	7
Training Set Size	4,478	13,084
Development Set Size	500	700
Testing Set Size	893	700

Table 2.1: Dataset Summary

Model	ATIS			SNIPS		
	Slot (F1)	Intent (ACC)	Sent (ACC)	Slot (F1)	Intent (ACC)	Sent (ACC)
Joint Sequence - Hakkani-Tür *et al.* (2016)	94.3	92.6	80.7	87.3	96.9	73.2
Attention Based - Liu and Lane (2016)	94.2	91.1	78.9	87.8	96.7	74.1
Slot-gate (Full Atten.) - Goo *et al.* (2018)	94.8	93.6	82.2	88.8	97.0	75.5
Slot-gate (Intent Atten.) - Goo *et al.* (2018)	95.2	94.1	82.6	88.3	96.8	74.6

Table 2.2: Evaluations on Previous Systems

Average data per class in ATIS (roughly 213 examples) is less than SNIPS Average data (roughly 1870 examples) as shown in Table 2.1. If you see table 2.2 all the models are able to perform better on SNIPS in terms of intent accuracy. So, there is possibility of data sparsity in ATIS.

When generating more data, it is important that we create relevant data, keeping context of the original data. So, it is very important that we find which sense of word is used in this context to generate similar sentences.

2.2 Word Sense Disambiguation

One of the famous problems in natural language processing is word sense disambiguation. A word in a sentence has a unique meaning given the context of the sentence. But if you look for word usage in a dictionary, a word can have more than one sense. Finding which dictionary sense is used in a particular sentence is an open problem and is termed as word sense disambiguation.
For example:

1. For a word "book", following is the dictionary meaning:

   (a) A written or printed work

(b) Reserve, buy in advance

(c) Make an official record of the name

Given a sentence: "*I want to book a flight from Phoenix to Boston*"

The (b) definition of word is used.

Given a sentence: "*I bought a book of selected resources*"

The (a) definition of word is used.

2. For a word "bank", you have dictionary meaning :

   (a) a long pile or heap; mass

   (b) the slope immediately bordering a stream course along which the water normally runs

   (c) an institution for receiving, lending, exchanging, and safeguarding money and, in some cases, issuing notes and transacting other financial business.

   (d) to form or group in a tier

   (e) to incline laterally

   Given a sentence: *"The bank will not be accepting any cash or check on Saturdays"*

   The (c) definition of word is used.

   Given a sentence: *"The river overflowed the bank"*

   The (b) definition of word is used.

If we see the examples above, we can understand that the same word can be used in different context. This is important for the paper because if we know the exact sense used in a training phase, we can extend our training data using synonyms of that particular sense to generate relevant sentences for the domain. There are various Word Sense Disambiguation algorithms :

1. **Thesaurus/Dictionary Methods**: In this method we use the dictionary definition of the various word(s) used in the sentence, to find out which sense is used in a particular sentence. One of the famous approaches for solving this is the Lesk algorithm. The Lesk algorithm is based on assumption that words in a given "neighborhood" will tend to share a common topic.
   The Algorithm of simplest Lesk is:

   (a) Find all sense definitions of the disambiguated word.

   (b) Determine the definition overlap for all possible sense combination

   (c) Choose senses that lead to highest overlap.

   An example showing working of a Lesk algorithm:
   Let's assume we need to disambiguate "PINE CONE":
   Definition of "PINE":

   (a) kinds of evergreen tree with needle-shaped leaves

   (b) waste away through sorrow or illness

   Definition of "CONE":

   (a) solid body which narrows to a point

   (b) something of this shape whether solid or hollow

   (c) fruit of certain evergreen tree

   The Lesk algorithm output is based on a similarity measure between different definitions. There are various ways in which this comparison can be carried out. There are various versions of Lesk algorithm based on similarity measure used: Original Lesk Lesk (1986), Adapted Lesk Banerjee and Pedersen (2002) and Enhanced Lesk Basile *et al.* (2014), etc.

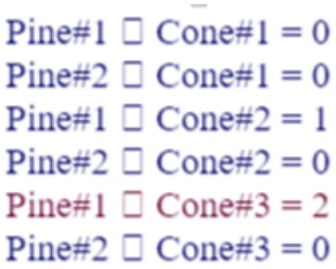


Figure 2.1: Cross Comparison of Dictionary Definitions of PINE and CONE

2. Machine learning approaches: Most of the learning approaches have been based on a supervised learning technique. Some of the famous ones are: using Support vector machines Lee *et al.* (2004), Zhong and Ng (2010).

We implemented word sense disambiguation library Tan (2014), as a sub-process to generate more relevant examples.

2.3 Data Augmentation Algorithm

As discussed earlier, one of the regularization techniques is data augmentation. It works on the concept that if we provide more relevant training data, we can expect a learning algorithm to generalize better. Then we say that we can use word sense disambiguation to find the sense of a word in a sentence. Using both these points, we can create a system where one can use word sense disambiguation to create relevant data and use this relevant data as a regularization technique in our learning algorithm to generalize better.

Algorithm for data augmentation with working example:

Example Sentence: *"i'm looking for a flight from charlotte to las vegas that stops in st. louis hopefully a dinner flight how can i find that out"*

1. Data preprocessing: This includes converting into sentences into lower form and performing lemmatization.

After this step : *"i am looking for a flight from charlotte to las vegas that stops in st. louis hopefully a dinner flight how can i find that out"*

2. Perform tokenization.

   After this step: ["i","am", "looking", "for", "a", "flight", "from", "charlotte", "to", "las", "vegas", "that", "stops", "in", "st.", "louis", "hopefully", "a", "dinner", "flight", "how", "can", "i", "find", "that", "out"]

3. Search for all the words which need to be disambiguated.

   After this step: ["looking", "flight", "charlotte", "vegas", "stops", "louis", "hopefully", "dinner", "flight", "find"]

4. Filter out all the words which are represented by "B" or "I" in the IOB format of slot annotation. The reason behind this is in future we want to learn their slot, so replacing them will not create a relevant sentence.

   The Annotated Slot: ["O", "O", "O", "O", "O", "O", "O", "$B-fromloc.city\_name$" , "O", "B-toloc.city_name", "I-toloc.city_name", "O", "O", "O", "B-stoploc.city_name", "I-stoploc.city_name", "O", "O", "B-meal_description", "O", "O", "O", "O", "O", "O", "O"]

   After this step: ["looking", "flight", "stops", "hopefully", "flight", "find"]

5. For each disambiguated word, use Lesk Algorithm.

   After this step: see table 2.3 on page 21.

6. For each disambiguated word, find synonyms and antonyms using its sense.

   After this step: see table 2.4 on page 21.

7. Create all possible combination of sentence using new words.

   After this step:

looking	Synset('search.v.02')
flight	Synset('trajectory.n.01')
stops	Synset('stop.v.05')
hopefully	Synset('hopefully.r.02')
find'	Synset('witness.v.02')

Table 2.3: Sysnet Output in WSD

Word	Synonyms	Dictionary Definition	Antonyms
looking	look, search	search or seek	back
flight	flight, trajectory	path followed by an object moving through space	N/A
stops	stop	cause to stop	begin, start
hopefully	hopefully	it is hoped	hopelessly
find	witness, see, find	perceive or be contemporaneous with	lose

Table 2.4: Extraction of Synonyms and Antonyms Based on Sense

(a) Similarity Data:

i. *"i am searching for a flight from charlotte to las vegas that stop in st. louis hopefully a dinner trajectory how can i witness that out"*

ii. *"i am looking for a trajectory from charlotte to las vegas that stop in st. louis hopefully a dinner flight how can i witness that out"*

iii. *"i am searching for a flight from charlotte to las vegas that stop in st. louis hopefully a dinner flight how can i witness that out"*

so on .....

(b) Dissimilarity Data:

i. *"i am looking for a flight from charlotte to las vegas that stop in st. louis hopelessly a dinner flight how can i lose that out*

ii. *"i am backing for a trajectory from charlotte to las vegas that start in st. louis hopefully a dinner flight how can i lose that out"*

iii. *"i am not looking for a flight from charlotte to las vegas that stop in st. louis hopefully a dinner flight how can i not find that out"*

so on .....

Similarity Data Examples	
ATIS	**SNIPS**
usher me the flight from dallas to baltimore in first class	tally slimm cutta calhoun to my this is prince playlist
show going and comer in atlanta for american airlines	is it probable to be warm in rush hill
what flight are usable from denver to san francisco	usher me the best of : volume 1 tv series
do i have a meal on the atlanta to bwi flight eastern 210	i had wish to know how the weather will be at 8 pm in tennessee
usher me the earliest nonstop flight from dallas to houston	how do i see the soundtrack african development perspectives yearbook
i would care to make a booking for a flight to denver from philadelphia on this coming sunday	i would wish to book a restaurant in poncha springs for 8 at 00:32 am

Table 2.5: Examples of Generated Similarity Data

Dissimilarity Data Examples	
ATIS	**SNIPS**
hide me the flights from dallas to baltimore in first class	please take away iris dement to my playlist this is selena
disprove departures and arrivals in atlanta for american airlines	is it improbable to be warm in rush hill
what flights are unavailable from denver to san francisco	hide me the best of : volume 1 tv series
do i take away a meal on the atlanta to bwi flight eastern 210	i had dislike to know how the weather will be at 8 pm in tennessee
disprove me the nonstop flights from dallas to houston	how do i lose the soundtrack african development perspectives yearbook
i would like to unmake a reservation for a flight to denver from philadelphia on this coming sunday	i would dislike to book a restaurant in poncha springs for 8 at 00:32 am

Table 2.6: Examples of Generated Dissimilarity Data

Chapter 3

LEARNING ON AUGMENTED DATA

3.1 Motivation for BiRNN

In intent classification problem, the data is in the form of natural language. In natural language, sequence in which you formulate the data changes the meaning and output of the data. So, we are looking for a learning model which can do processing on sequential data.

In sequential data, a traditional fully connected feed-forward network would need to learn separate parameters for each input feature. It would need to learn all the rules of the language separately at each position in the sentence. Whereas the parameter sharing is different in recurrent neural networks. At each step, the new output includes a function upon the previous output. Each member of the output is produced using same update rule applied to the previous output.

Looking at the NLU system (Natural language understanding system), we want a model to learn two tasks, intent classification and slot filling. In Intent classification, we are trying to learn a Many-to-One function, where given a sequence of n words, we are trying to learn a high level idea or context of the sentence.

In slot filling, we are trying to learn Many-to-Many function, where for each word in a sequence we are trying to learn if it belongs to either of "I", "O" or "B" chunk. These chunks are based on IOB format, where "I" means inside a chunk, "O" means outside a chunk and "B" means beginning of a chunk. [1]

Looking into sequential pattern-finding properties of RNN (Recurrent Neural Net-

[1] `https://www.deeplearningbook.org/contents/rnn.html`

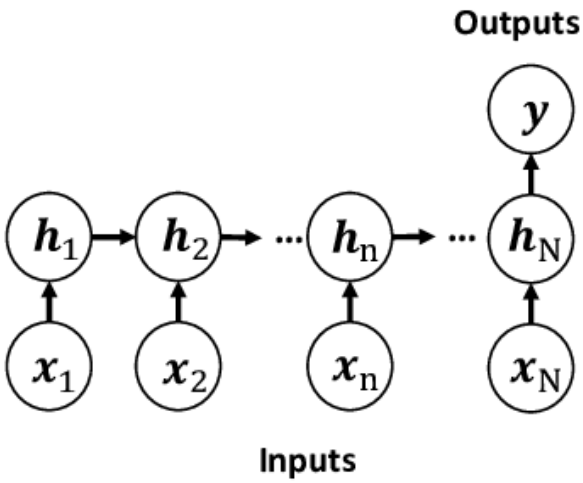


Figure 3.1: Many-to-One Recurrent Neural Network

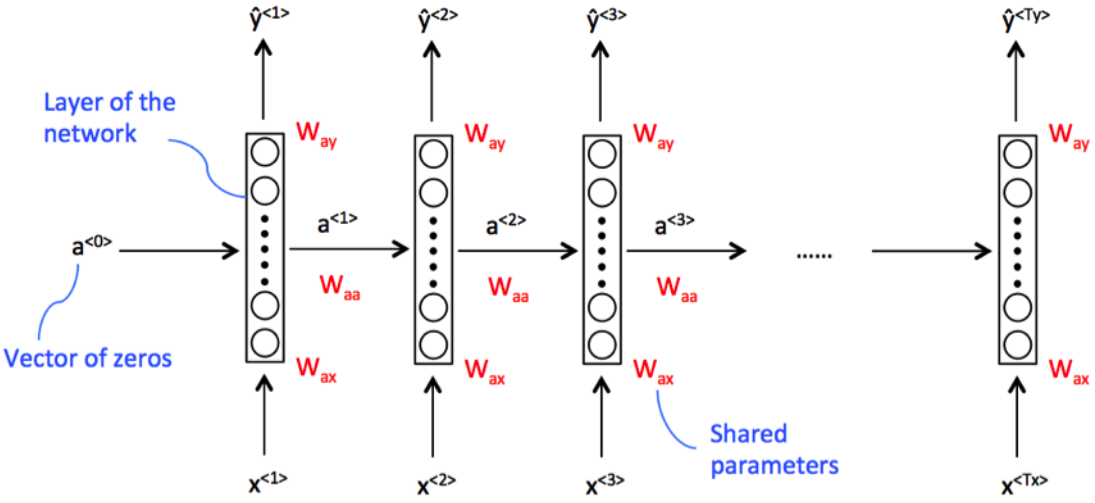


Figure 3.2: Many-to-Many Recurrent Neural Network

work), it seems like a good fit for the data. The figure 3.1 on page 26 shows a RNN model overview based on Many-to-One functionality which we can use for intent classification. For slot filling, figure 3.2 on page 26 shows a RNN model overview based on Many-to-Many functionality which can possibly be used.

A word in natural language is not necessarily only dependent upon the previously used words in sentences but also on the words which follows it in that sentence. Is there a way that a word can be represented as a vector which considers the context in both directions? For a word w_i in a sentence of n words, can we represent the w_i using two vectors, forward and backward. The forward vector will represent state from w_0 to w_i and backward will represent a vector from w_n to w_i. A bidirectional Recurrent neural network is a variant of RNN, where rather than taking into account only past states, every state is represented by two hidden states vectors, one for past states and other for future states. So, seeing this property of BiRNN , it seems like we can use BiRNN for our application. Most of the work in previous few years have been based on learning by BiRNN.

3.2 Objective Function

In this thesis, we want to improve intent classification and slot filling. The objective function in this can be viewed as

$$P(y^{Slot}, y^{intent}|x) \tag{3.1}$$

where x represent a given sequence of n words. As slot filling is a prediction task for each word in x in category: "I","O" or "B". Intent prediction is a prediction task given a sequence of n words. So the objective function can be formulated as:

$$p(y^{Intent}|x) \prod_{i=1}^{n} p(y_i^{Slot}|x) \tag{3.2}$$

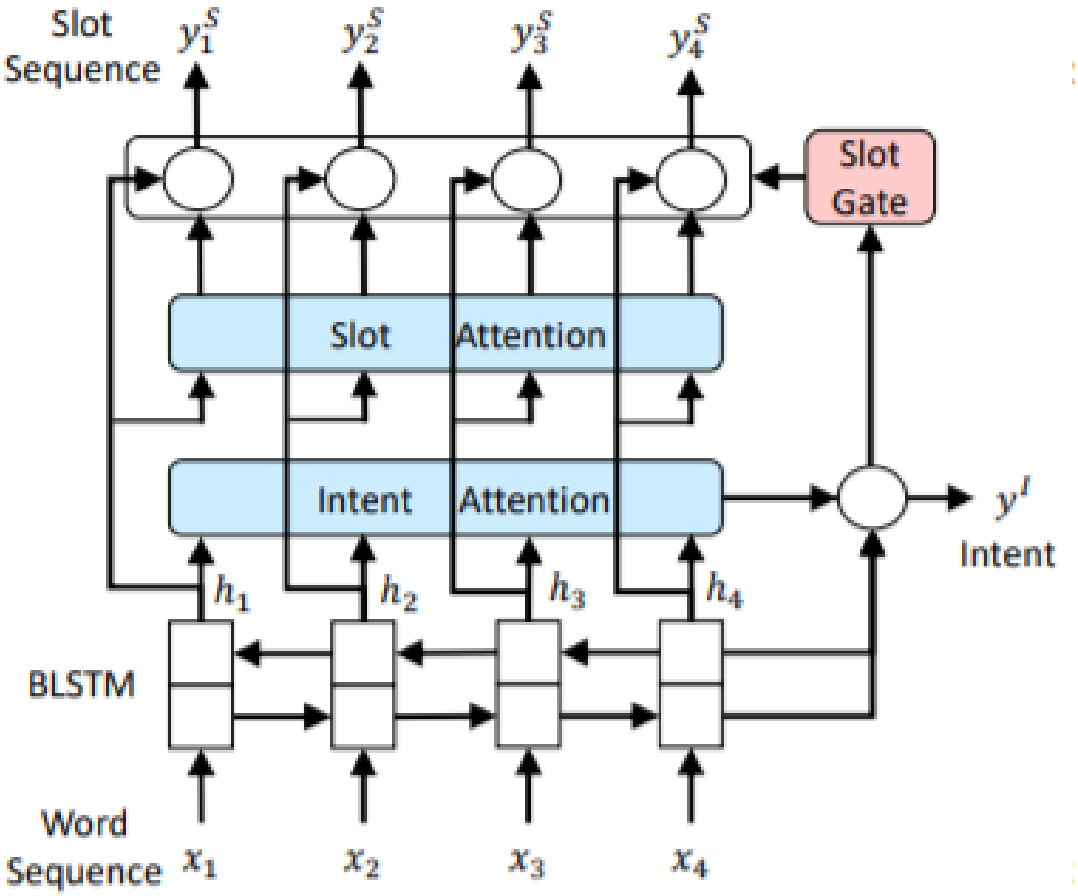


Figure 3.3: Slot-Gated Model with Full Attention

Goo *et al.* (2018)

$$P(y^{Intent}|x_1, x_2, ...x_n) \prod_{i=1}^{n} p(y_i^{Slot}|x_1, x_2, ...x_n) \tag{3.3}$$

3.3 Step-By-Step Model Architecture

In this section, we will explain attention-based RNN model used in this paper.

1. **Creating hidden states**: The bidirectional long short-term memory (BLSTM) model, Mesnil *et al.* (2015), takes a word sequence x = (x_1, . . . , x_T) as input,

and then generates forward hidden state $\vec{h_i}$ and backward hidden state $\overleftarrow{h_i}$. A state at a time i is represented by [h_i =$\vec{h_i}$, $\overleftarrow{h_i}$]

2. For **Slot filling**: x is mapping to its corresponding slot label sequence y =(y_1^S,..., y_T^S). For each hidden state h_i , we compute the slot context vector c_i^S as the weighted sum of LSTM's hidden states, h_1, ..., h_T ,by the learned attention weights $\alpha_{i,j}^S$:

$$c_i^S = \sum_{j=1}^{T} \alpha_{i,j}^S h_j \tag{3.4}$$

where the slot attention weights are computed as below.

$$\alpha_{i,j}^S = \frac{exp(e_{i,j})}{\sum_{k=1}^{T} exp(e_{i,k})} \tag{3.5}$$

$$e_{i,k} = \sigma(We_{he}^S h_k) \tag{3.6}$$

where σ is the activation function, and W_{he}^S he is the weight matrix of a feed-forward neural network.Then the hidden state and the slot context vector are utilized for slot filling:

$$y_i^S = softmax(W_{hy}^S(h_i + c_i^S)) \tag{3.7}$$

where y_i^S is the slot label of the i-th word in the input,and W_{hy}^S is the weight matrix.

3. For **Intent Prediction**: The intent context vector c^I can be computed in the same manner as c^S, but the intent detection part only takes the last hidden state of BLSTM. The intent prediction is:

$$y^I = softmax(W_{hy}^I(h_T + c^I)) \tag{3.8}$$

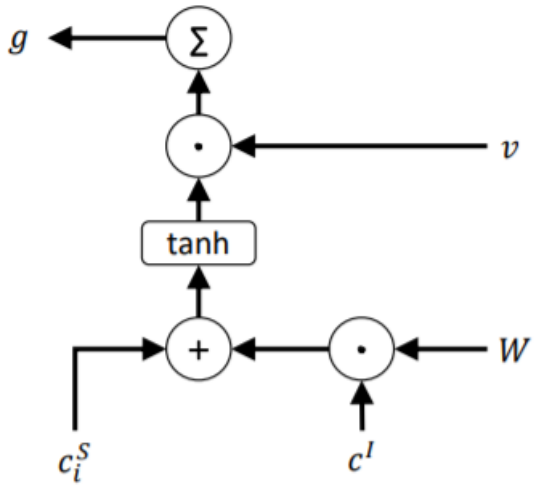


Figure 3.4: Slot Gate Illustration

Goo *et al.* (2018)

4. **Slotted Gate Mechanism**: We first combine slot context vector c_i^S and intent context vector c_I to pass through a slot gate.

$$g = \sum v.tanh(c_i^S + W * c^I) \tag{3.9}$$

where v and W are trainable vector and matrix respectively.

We use g to weight between h_i and c_i^S to derive y_i^S as below:

$$y_i^S = softmax(W_{hy}^S(h_i + c_i^S * g)) \tag{3.10}$$

where g is slot context vector.

Chapter 4

EXPERIMENTS AND RESULTS

4.1 Data Set

For evaluation we used two data sets, ATIS and SNIPS. The ATIS (Airline Travel Information Systems) dataset Tur *et al.* (2010) is commonly used in Spoken Language Understanding System. The data is constructed from voice conversations of people making flight reservations. The number of training utterances are 4478 and test contains 893 utterances. The example sentences of ATIS can be observed in table 4.3 on page 35.

We also tested our approach on SNIPS dataset. This data set is collected from the Snips personal voice assistant [1] . The number of training utterances in SNIPS are 13,084 and test contains 700 utterances. The example sentences of SNIPS can be observed in table 4.2 on page 32

Points to Note:

1. While ATIS covers different scenarios in single domain of airline data, the SNIPS dataset covers 7 different domains.

2. The vocabulary size of SNIPS (11,241) is significantly large as compared to ATIS (722).

3. Moreover, intents in ATIS are unbalanced, "atis_flight" class accounts for about 74% of the training data. This can be seen in figure 4.1 on 36. The class distribution in SNIPS is balanced as shown in figure 4.2 on 37.

[1] `https://github.com/snipsco/nlu-benchmark/tree/master/2017-06-custom-intent-engines`

Dataset Name	ATIS	SNIPS
Vocabulary Size	722	11,241
Number of slots	120	72
Number of Intents	21	7
Training Set Size	4,478	13,084
Development Set Size	500	700
Testing Set Size	893	700

Table 4.1: Dataset Details

Intent	Utterance Example
SearchCreativeWork	Find me the I, Robot television show
GetWeather	Is it windy in Boston, MA right now?
BookRestaurant	I want to book a highly rated restaurant tomorrow night
PlayMusic	Play the last track from Beyonc off Spotify
AddToPlaylist	Add Diamonds to my roadtrip playlist
RateBook	Give 6 stars to Of Mice and Men
SearchScreeningEvent	Check the showtimes for Wonder Woman in Paris

Table 4.2: One Training Sample of Each Class in SNIPS Dataset

All the details of ATIS and SNIPS can be seen in table 4.1 on page 32. The working of Natural language Understanding System can be understood as a system which, first given a utterance, finds best domain, then finds best class in that domain. SNIPS can be seen as covering first scenario and ATIS can be seen as covering in depth working of a single domain.

Intent Class	Example
$atis\_airline\#atis\_flight\_no$	may i please see airlines and flight numbers from new york to toronto on the same date june seventeenth also arriving in toronto before noon thank you
$atis\_capacity$	what are the seating capacities of planes between pittsburgh and baltimore
$atis\_quantity$	how many flights does twa have in business class
$atis\_airfare$	round trip fares from baltimore to philadelphia less than 1000 dollars round trip fares from denver to philadelphia less than 1000 dollars round trip fares from pittsburgh to philadelphia less than 1000 dollars
$atis\_ground\_service\#atis\_ground\_fare$	what ground transportation is available from the pittsburgh airport to downtown and how much does it cost
$atis\_distance$	how long does it take to get from atlanta airport into the city of atlanta
$atis\_ground\_fare$	what are the rental car rates in san francisco

$atis\_flight\_no$	list the number of flights arriving in dallas fort worth from boston before noon
$atis\_ground\_service$	what types of ground transportation are there to san francisco airport
$atis\_airport$	houston airports
$atis\_flight\#atis\_airfare$	please give me a list of all the flights between dallas and baltimore and their cost
$atis\_cheapest$	show me the cheapest fare in the database
$atis\_flight$	i want to fly from baltimore to dallas round trip
$atis\_abbreviation$	what does fare code y mean
$atis\_restriction$	what is restriction ap57
$atis\_aircraft\#atis\_flight\#atis\_flight\_no$	i want to fly from detroit to st. petersburg on northwest airlines and leave around 9 am tell me what aircraft are used by this flight and tell me the flight number
$atis\_airline$	which airlines fly from boston to washington dc via other cities
$atis\_meal$	do i get a meal on the atlanta to bwi flight eastern 210

$atis\_aircraft$	okay i would like to know the type of aircraft used on a flight from cleveland to dallas please
$atis\_flight\_time$	could you give me the schedule of flights for american and delta to dfw on august fifteenth
$atis\_city$	what time zone is denver in

Table 4.3: One Training Sample of Each Class in ATIS Dataset

4.1.1 Input Data Representation

The input data contains three parameters Word Sequence(W), Annotated Data(Slots), Class(Intent).

1. An example from ATIS:
   (a) Word Sequence(W): *show me all the flights from philadelphia to cincinnati*
   (b) Annotated Data(Slot): *O O O O O O B-fromloc.city_name O B-toloc.city_name*
   (c) Class(Intent): atis_flight
2. An example from SNIPS:
   (a) Word Sequence(W): *find me showtimes for animated movies in the neighbourhood*
   (b) Annotated Data(Slot): *O O O O B-movie_type I-movie_type B-spatial_relation I-spatial_relation I-spatial_relation*
   (c) Class(Intent): SearchScreeningEvent

Intent Class	# of Sample	% contribution
atis_flight	3309	73.89
atis_airfare	385	8.6
atis_ground_service	230	5.14
atis_airline	139	3.1
atis_abbreviation	130	2.9
atis_aircraft	70	1.56
atis_flight_time	45	1
atis_quantity	41	0.92
atis_flightatis_airfare	19	0.42
atis_city	18	0.4
atis_distance	17	0.38
atis_airport	17	0.38
atis_capacity	15	0.33
atis_ground_fare	15	0.33
atis_flight_no	12	0.27
atis_meal	6	0.13
atis_restriction	5	0.11
atis_airlineatis_flight_no	2	0.04
atis_aircraftatis_flightatis_flight_no	1	0.02
atis_ground_serviceatis_ground_fare	1	0.02
atis_cheapest	1	0.02

Figure 4.1: ATIS dataset sample distribution for each class

Intent Class	# of Sample	% contribution
AddToPlayList	1818	13.89483338
SearchScreeningEvent	1852	14.15469275
PlayMusic	1914	14.62855396
BookRestaurant	1881	14.37633751
SearchCreativeWork	1847	14.11647814
RateBook	1876	14.3381229
GetWeather	1896	14.49098135

Figure 4.2: SNIPS dataset sample distribution for each class

In the ATIS example above, the length of the Annotated Data and Word Sequence is same. It is valid for all the samples because for Annotated data(Slots), we are learning a one-to-one mapping. The slots are annotated in IOB format, where "I" means Inside chunk, "O" means outside of chunk and "B" means beginning of chunk.

1. In ATIS example, {"show", "me", "all", "the", "flights", "from" , "to"} are outside the chunk.
2. In ATIS example, {"philadelphia", "cincinnati"} are beginning the chunk.
3. In SNIPS example, {"the", "neighbourhood"} are inside the chunk, begin by "in".

4.2 Experimental Setup

We are defining three terms here, which will be used to explain experimental setup.

1. **X** represents the original training data.
2. $\mathbf{X}_{similar}$ represents the augmented similar sentences which are automatically generated using word sense disambiguation.
3. $\mathbf{X}_{dissimilar}$ represents the new dissimilar sentences which are automatically generated using word sense disambiguation.

Defining different types of Accuracy:

1. **Slot F1**: The slot F1 accuracy is based on how many slots you are able to identify(Recall) and how many of them are correct(Precision).
2. **Intent Accuracy**: If you classify intent correct, you get one point, else you get zero.

3. **Semantic Accuracy**: If you correctly identify intent of the user utterance along with all the slots inside it, then you get one point else you get zero.

4.2.1 Impact of Scaling on Data

In this experiment, we are trying to show that it's hard for a human generated training data to cover all the scenarios. For this experiment, we will show results using different scale of augmentation on training data.

Defining Scaling : The scaling rule is to generate maximum number of different combinations possible of disambiguated words in a user utterance. It can be any number n (ranging from 0 to N), where N is sufficiently large number. Let us suppose we want to scale by factor s, If s $<$ N, then we randomly take out s samples from N and add them as new similarity sentences. If s $>=$ N, then we add all generated sentences as new similarity sentences.

Objective function for this experiment is:

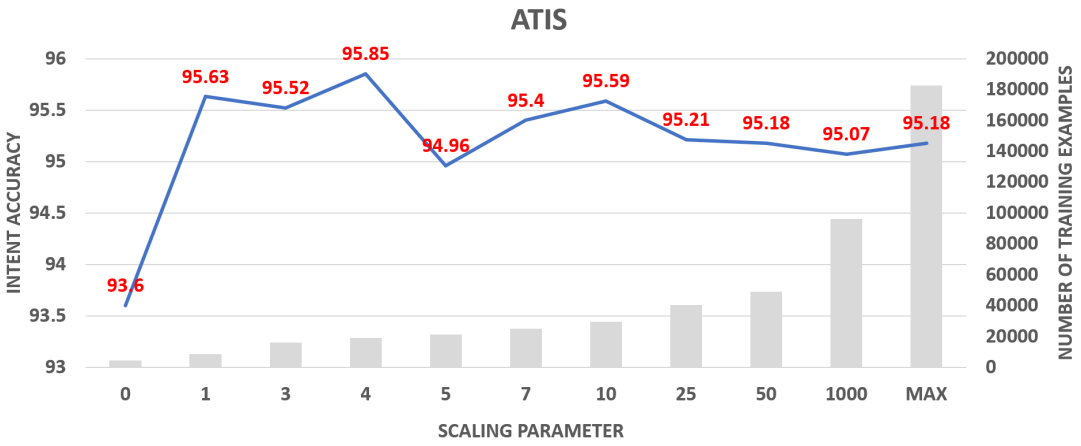
$$P(y^{Slot}, y^{intent}|X, X_{similar}) \tag{4.1}$$

Observations on ATIS dataset:

1. **Slot F1**: We observe in figure 4.4 on page 41 that our data augmentation technique is able to generate more accurate results on slot F1, our results are in range 95.13 (maximum data) to 95.85(scaling factor 5).

2. **Intent Accuracy**: We observe in figure 4.3 on page 40 that our data augmentation technique is able to generate better results on intent accuracy and the graph shows our results lie between range 94.96 to 95.85. We have shown significant improvement on previous state of the art.

3. **Semantic Accuracy**: We observe in figure 4.5 on page 41 that the results are overall better but there is no trend.



Model	ATIS DATASET
Joint Seq. (Hakkani-Tur et al,2016)	92.6
Atten.-Based (Liu and Lane, 2016)	91.1
Slot-gated (Goo et al. , 2018)	93.6

Figure 4.3: ATIS: Scaling VS Intent

Observations on SNIPS dataset:

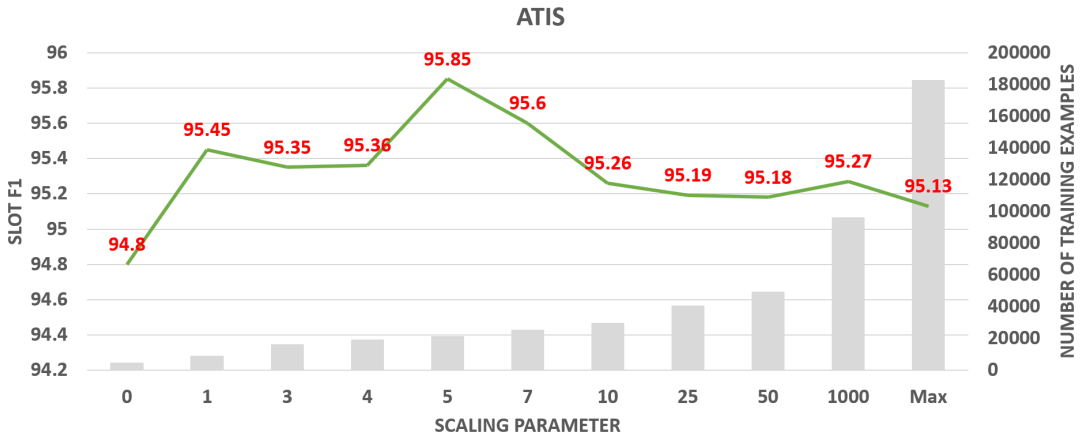
1. **Slot F1**: We observe in figure 4.9 on page 43 that our data augmentation technique is able to generate more accurate results on slot F1. we observe that after a scaling factor of 25, slot f1 is tending to stabilize.

2. **Intent Accuracy**: We observe in figure 4.8 on page 43 that our data augmentation technique is able to generate better results on intent accuracy and the graph shows similar trend as observed in slot f1.

3. **Semantic Accuracy**: We observe in figure 4.10 on page 41 that the results are overall better. After the scale factor of 50, it tends to generalize at 78.3 .

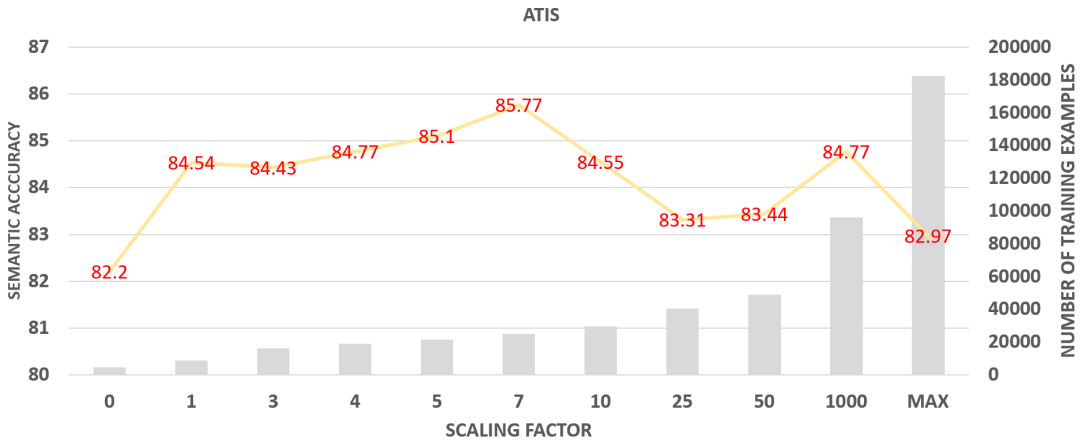
4.2.2 Impact of Negation Semantics

We are defining a new way of looking at intent accuracy, inspired by F1 score. Rather than looking at binary 1 for correct and 0 for incorrect, we want a system



Model	ATIS DATASET
Joint Seq. (Hakkani-Tur et al,2016)	94.3
Atten.-Based (Liu and Lane, 2016)	94.2
Slot-gated (Goo et al. , 2018)	94.8

Figure 4.4: ATIS: Scaling VS Slot F1



Model	ATIS DATASET [Semantic Score]
Joint Seq. (Hakkani-Tur et al,2016)	80.7
Atten.-Based (Liu and Lane, 2016)	78.9
Slot-gated (Goo et al. , 2018)	82.2

Figure 4.5: ATIS: Scaling VS Semantic

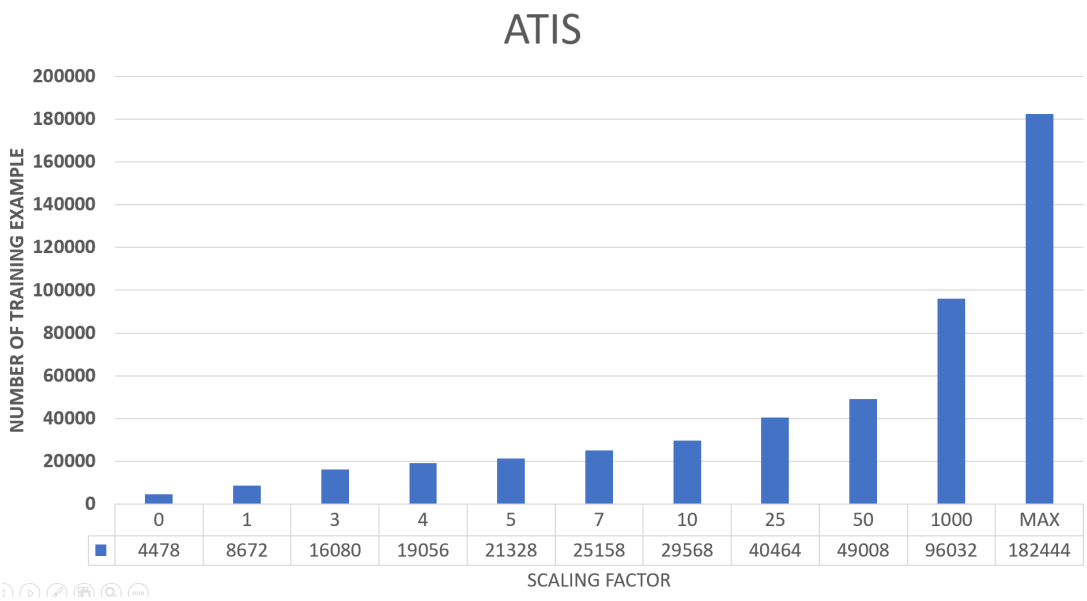


Figure 4.6: ATIS: Scaling VS Number of Training Samples

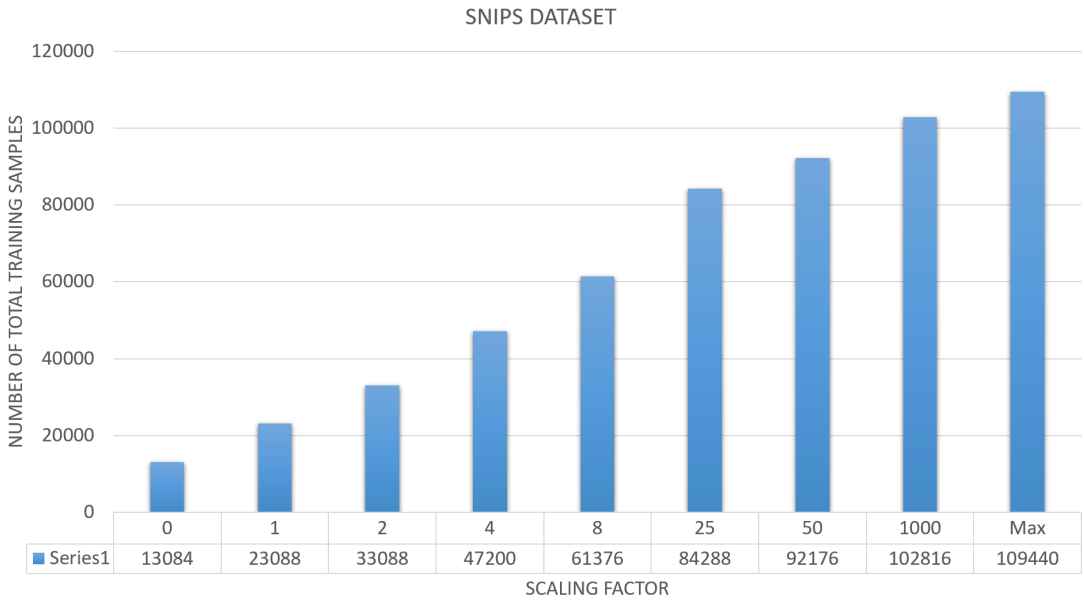
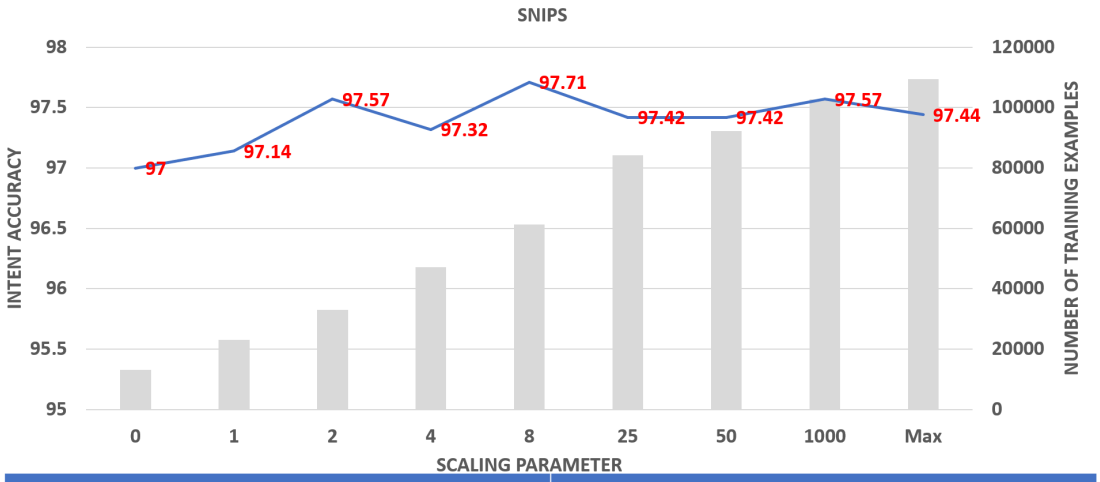
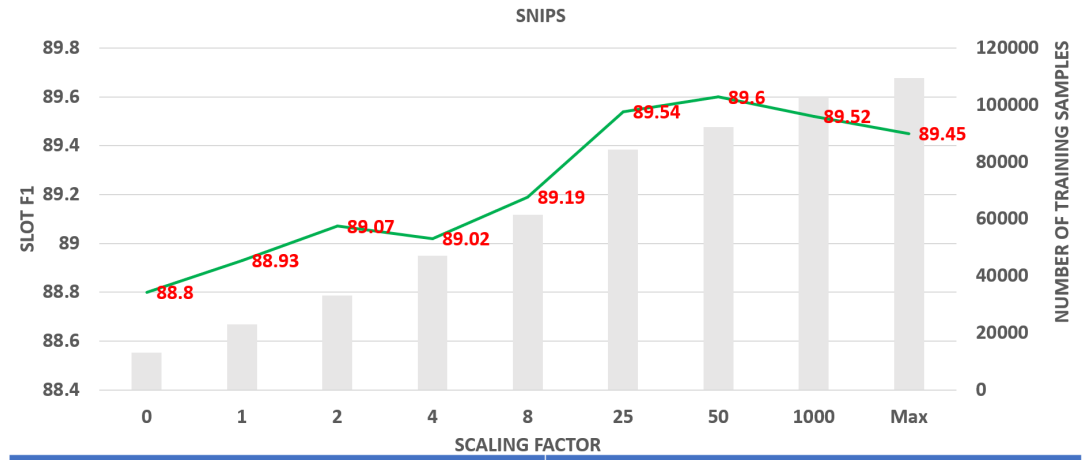


Figure 4.7: SNIPS: Scaling VS Number of Training Samples



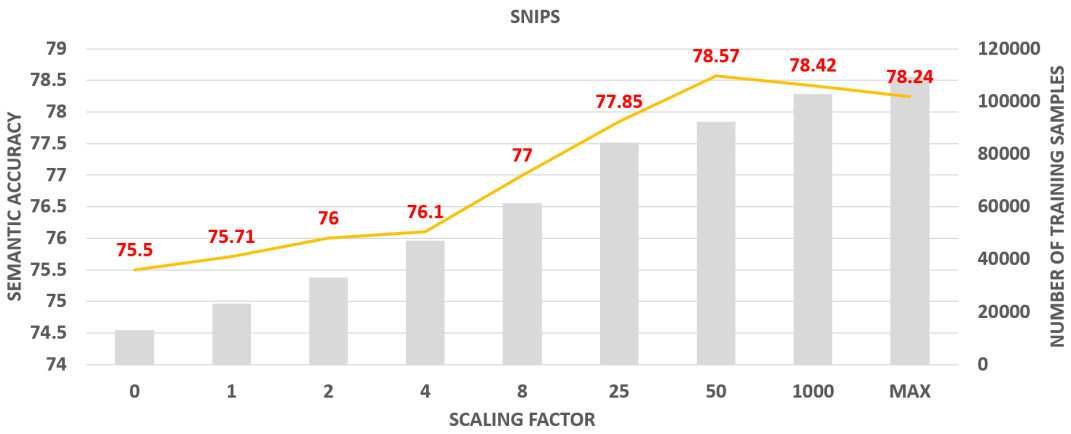
Model	SNIPS DATASET
Joint Seq. (Hakkani-Tur et al,2016)	96.9
Atten.-Based (Liu and Lane, 2016)	96.7
Slot-gated (Goo et al. , 2018)	97.0

Figure 4.8: SNIPS: Scaling VS Intent



Model	SNIPS DATASET
Joint Seq. (Hakkani-Tur et al,2016)	87.3
Atten.-Based (Liu and Lane, 2016)	87.8
Slot-gated (Goo et al. , 2018)	88.8

Figure 4.9: SNIPS: Scaling VS Slot F1



Model	SNIPS DATASET
Joint Seq. (Hakkani-Tur et al,2016)	73.2
Atten.-Based (Liu and Lane, 2016)	74.1
Slot-gated (Goo et al. , 2018)	75.5

Figure 4.10: SNIPS: Scaling VS Semantic

which can account negative marking for incorrect. We want to create a system which is more robust to adversarial examples and wrong classification. The intention is to develop a way to incorporate risk associated with misclassification. This is based on the idea that "it's better to have no knowledge than wrong knowledge". All the previous state of the art systems can be seen as shown in table 4.4 to 4.7 .

In this setup we are doing 2 type of experiments:

1. **Adding only dissimilar sentences**: We want to see the impact of adding negation semantics in original data. The objective function is:

$$P(y^{Slot}, y^{intent}|X, X_{dissimilar}) \tag{4.2}$$

The ATIS dataset is trained using all the original training data and randomly chosen 200 samples from dissimilar sentences. The SNIPS dataset is trained with all the original training data and randomly chosen 1900 samples from dissimilar sentences. The reason of choosing 200 and 1900 samples for ATIS

and SNIPS respectively, is based on the average amount of data per class in the data set.

The results for ATIS can be seen in table 4.4 on page 46 and for SNIPS can be seen in table 4.5 on page 47.

2. **Adding dissimilar and similar sentences sentences** : The objective function is:

$$P(y^{Slot}, y^{intent}|X, X_{dissimilar}, X_{similar}) \tag{4.3}$$

The ATIS dataset is trained using all the original training data (X) , similar data generated with scaling factor of 4 ($X_{similar}$) and randomly chosen 900 samples from dissimilar sentences ($X_{dissimilar}$) . The SNIPS dataset is trained with all the original training data (X) , similar data generated with scaling factor of 8 ($X_{similar}$) and randomly chosen 9000 samples from dissimilar sentences ($X_{dissimilar}$) .The reason for augmenting the similarity data ($X_{similar}$) by 4 and 8 times for ATIS and SNIPS respectively, is based on best intent accuracy results found in earlier experiment of impact of scaling, as shown in graph 4.3 on page 40 for ATIS and graph 4.8 on page 43 for SNIPS. The reason for choosing 900 and 9000 samples for ATIS and SNIPS respectively, is based on the average amount of data per class in the data set (After performing data augmentation based on similarity).

Observations on ATIS dataset:

1. $\mathbf{X} + \mathbf{X}_{dissimilar}$: In table 4.4 on page 46, we observed that the recall is 99.45, which means that for 5 samples in the test data, model does not know the intent category. This resulted in getting higher precision value and overall F1 score is better than any of three state of the art system intent F1 value.

	Recall	Precision	F1 Score
Joint Sequence - Hakkani-Tür *et al.* (2016)	100	92.6	96.2
Attention Based - Liu and Lane (2016)	100	91.1	95.3
Slot-gate (Full Atten.) - Goo *et al.* (2018)	100	93.6	96.7
Our Approach	99.45	95.38	**97.37**

Table 4.4: Result on ATIS using X + $X_{dissimilar}$

2. $\mathbf{X} + \mathbf{X}_{similar} + \mathbf{X}_{dissimilar}$: In table 4.6 on page 47, we observed that recall is 99.21, which means for 7 samples in the test data, model does not know the intent category. This resulted in getting higher precision value and overall F1 score is better than any of three state of the art system intent F1 value.

Observations on SNIPS dataset:

1. $\mathbf{X} + \mathbf{X}_{dissimilar}$: In table 4.5 on page 47, we observed that our recall is 99.85, which means that for 1 sample in the test data, model does not know the intent category. This resulted in getting higher precision value.

2. $\mathbf{X} + \mathbf{X}_{similar} + \mathbf{X}_{dissimilar}$: In table 4.7 on page 47, we observed that our recall is 99.57, which means that for 3 samples in the test data, model don't know the intent category. This shows precision value has gone up and F1 score is 98.40. This F1 score does not outperform the Goo *et al.* (2018) performance.

4.3 Analysis of Results

In this section, we will do analysis on all the experiments we did.

1. **Analysis on X** : we need to first look into the errors generated on previous state of the art system to understand where we are doing better and where we

	Recall	Precision	F1 Score
Joint Sequence - Hakkani-Tür *et al.* (2016)	100	96.9	98.40
Attention Based - Liu and Lane (2016)	100	96.7	98.30
Slot-gate (Full Atten.) - Goo *et al.* (2018)	100	97.0	98.47
Our Approach	99.85	97.42	**98.62**

Table 4.5: Result on SNIPS using X + $X_{dissimilar}$

	Recall	Precision	F1 Score
Joint Sequence - Hakkani-Tür *et al.* (2016)	100	92.6	96.2
Attention Based - Liu and Lane (2016)	100	91.1	95.3
Slot-gate (Full Atten.) - Goo *et al.* (2018)	100	93.6	96.7
Our Approach	99.21	96.28	**97.72**

Table 4.6: Result on ATIS using X + $X_{similar}$ $X_{dissimilar}$

	Recall	Precision	F1 Score
Joint Sequence - Hakkani-Tür *et al.* (2016)	100	96.9	98.40
Attention Based - Liu and Lane (2016)	100	96.7	98.30
Slot-gate (Full Atten.) - Goo *et al.* (2018)	100	97.0	98.47
Our Approach	99.57	97.27	**98.40**

Table 4.7: Result on SNIPS using X + $X_{similar}$ + $X_{dissimilar}$

need to improve.

(a) ATIS: In table 4.8 of misclassifications on ATIS using original training data (X), we observe that there are total 56 misclassifications.

i. There are 5 samples numbered 34, 36, 229, 492 and 499 which are misclassified, classes for these 5 samples are not even present in the training set. The dataset is expecting to learn the class name without any example in training.

ii. There are 14 samples (12, 206, 207, 208, 213, 229, 405, 406, 407, 492, 497, 498, 499 and 604) which have multiple intents and are misclassified. A point to observe is that intent classifier is able to predict one intent out of many but fails to recognize multiple intents.

iii. There are 4 samples (688, 689, 691 and 692) which have single intent and are misclassified for having multiple intents.

iv. There are 35 samples in which learning is not proper.

No	Actual Class	Predicted Class	Testing Utterance
12	atis_flight #atis_airfare	atis_flight	show flight and prices kansas city to chicago on next wednesday arriving in chicago by 7 pm
32	atis_ground _service	atis_distance	does tacoma airport offer transportation from the airport to the downtown area
34	atis_day_name	atis_flight_time	what day of the week do flights from nashville to tacoma fly on
36	atis_day_name	atis_flight	what days of the week do flights from san jose to nashville fly on

41	atis_flight	atis_abbreviation	does this flight serve dinner
50	atis_meal	atis_flight_time	what meals are served on american flight 811 from tampa to milwaukee
51	atis_meal	atis_flight_time	what meals are served on american flight 665 673 from milwaukee to seattle
93	atis_airport	atis_ground _service	i would like to know what airports are in los angeles
94	atis_flight	atis_ground _service	does the airport at burbank have a flight that comes in from kansas city
108	atis_city	atis_flight	to what cities from boston does america west fly first class
137	atis_ground_fare	atis_airfare	what's the fare for a taxi to denver
138	atis_ground_fare	atis_ground _service	what are the fares for ground transportation in denver
154	atis_airline	atis_distance	what airlines off from love field between 6 and 10 am on june sixth
176	atis_meal	atis_flight_time	what meals are available on dl 468 which al arrives in san francisco at 950 am
206	atis_flight #atis_airfare	atis_flight	list all flights and their fares from indianapolis to memphis on a monday morning
207	atis_flight #atis_airfare	atis_flight	list all flights and their fares from memphis to miami on a wednesday evening
208	atis_flight #atis_airfare	atis_airfare	list all flights and their fares for all flights between miami and indianapolis

213	atis_flight #atis_airfare	atis_flight	list all sunday flights from cleveland to nashville and their fares
216	atis_distance	atis_flight	how long does a flight from baltimore to san francisco take
229	atis_airfare #atis_flight	atis_airfare	list the airfare for american airlines flight 19 from jfk to lax
329	atis_meal	atis_flight_time	what meals are there on flight 382 from milwaukee to washington dc on tuesday morning
405	atis_flight #atis_airfare	atis_flight	i need a round trip flight from san diego to washington dc and the fares
406	atis_flight #atis_airfare	atis_flight	i need a round trip from atlanta to washington dc and the fares leaving in the morning
407	atis_flight #atis_airfare	atis_flight	i need a round trip from phoenix to washington dc and the fare leaving in the morning
469	atis_airfare	atis_flight	get the saturday fare from washington to toronto
492	atis_flight #atis_airline	atis_flight	i need flight and airline information for a flight from denver to salt lake city on monday departing after 5 pm
497	atis_flight #atis_airfare	atis_flight	i need flight and fare information for thursday departing prior to 9 am from oakland going to salt lake city
498	atis_flight #atis_airfare	atis_airfare	i need flight and fare information departing from oakland to salt lake city on thursday before 8 am

499	atis_flight_no #atis_airline	atis_flight_no	i need flight numbers and airlines for flights departing from oakland to salt lake city on thursday departing before 8 am
500	atis_flight_no	atis_flight_time	i need flight numbers for those flights departing on thursday before 8 am from oakland going to salt lake city
501	atis_airport	atis_flight	list airports in arizona nevada and california please
502	atis_airport	atis_flight	list california nevada arizona airports
528	atis_flight_no	atis_flight	i need the flight numbers of flights leaving from cleveland and arriving at dallas
604	atis_flight #atis_airfare	atis_flight	show me the nonstop flights and fares from toronto to st. petersburg
629	atis_city	atis_flight	list la
630	atis_city	atis_flight	list la
645	atis_airfare	atis_flight	list airfares for first class round trip from detroit to st. petersburg
654	atis_capacity	atis_flight	list seating capacities of delta flights from seattle to salt lake city
663	atis_flight	atis_airfare	give me the flights and fares for a trip to cleveland from miami on wednesday
670	atis_flight	atis_flight_time	give me the meal flights departing early saturday morning from chicago to seattle nonstop

688	atis_flight_no	atis_airline #atis_flight_no	flight number from dallas to houston
689	atis_flight_no	atis_airline #atis_flight_no	flight number from houston to dallas
691	atis_flight_no	atis_airline #atis_flight_no	flight numbers on american airlines from phoenix to milwaukee
692	atis_flight_no	atis_airline #atis_flight_no	flight numbers from chicago to seattle
722	atis_flight	atis_quantity	how many northwest flights leave st. paul
723	atis_flight	atis_quantity	how many northwest flights leave washington dc
724	atis_flight	atis_quantity	how many flights does northwest have leaving dulles
725	atis_city	atis_abbreviation	what cities does northwest fly out of
735	atis_airport	atis_city	show me the airports serviced by tower air
739	atis_meal	atis_city	are meals ever served on tower air
740	atis_meal	atis_flight	are snacks served on tower air
773	atis_capacity	atis_quantity	how many people will a 757 hold
801	atis_flight	atis_quantity	how many flights does alaska airlines have to burbank
881	atis_airline	atis_flight	is there one airline that flies from burbank to milwaukee milwaukee to st. louis and from st. louis to burbank
885	atis_airport	atis_flight	tell me all the airports near westchester county

887	atis_airport	atis_ground _service	tell me all the airports in the new york city area

Table 4.8: ATIS Misclassifications using Original X

(b) SNIPS: In Table 4.9 of misclassifications on SNIPS using original training data (X), we observe that there are total 22 misclassifications.

i. 15 of them are related to misclassification of "SearchCreativeWork" and "SearchScreeningEvent". Out of 15, 10 of them are in between these two classes.

ii. Samples like number 51 and 266 are hard to classify correctly even by humans.

No	Actual Class	Predicted Class	Testing Utterance
27	SearchScreeningEvent	RateBook	in one hour find king of hearts
51	SearchCreativeWork	PlayMusic	play the new noise theology e p
86	SearchCreativeWork	SearchScreeningEvent	where can i find conduct unbecoming
90	SearchScreeningEvent	SearchCreativeWork	find now and forever
110	SearchScreeningEvent	PlayMusic	what time will paris by night aired
121	BookRestaurant	SearchScreeningEvent	find a reservation at fish express
266	GetWeather	BookRestaurant	i need a table in uruguay in 213 days when it's chillier
345	SearchScreeningEvent	SearchCreativeWork	need to see mother joan of the angels in one second

408	SearchScreeningEvent	SearchCreativeWork	i'd like to watch take this waltz
426	PlayMusic	SearchCreativeWork	play the song shine a light
434	RateBook	SearchCreativeWork	the book history by contract is rated five stars in my opinion
517	SearchCreativeWork	PlayMusic	listen to dragon ball: music collection
528	SearchCreativeWork	SearchScreeningEvent	pull up sweeney todd - il diabolico barbiere di fleet street
545	SearchScreeningEvent	SearchCreativeWork	i'd like to watch wish you were dead
558	SearchScreeningEvent	GetWeather	what is the showtime for arsho
567	PlayMusic	SearchCreativeWork	play the album how insensitive
582	GetWeather	BookRestaurant	i need the wather for next week in the philippines
592	SearchScreeningEvent	SearchCreativeWork	find the panic in needle park
614	SearchCreativeWork	SearchScreeningEvent	i'm looking for circus world
664	SearchScreeningEvent	SearchCreativeWork	where can i see a slice of life
672	SearchCreativeWork	SearchScreeningEvent	show me the movie operetta for the theatre organ

Table 4.9: SNIPS Misclassifications using Original X

2. **Analysis on $\mathbf{X} + \mathbf{X}_{similar}$:**

   (a) ATIS: After looking at the graphs of 3 accuracy measures on ATIS, i.e graph 4.3, graph 4.4 and graph 4.5, we observed that there are peaks

coming near scaling 5. This is because the dataset is highly unbalanced (figure 4.1), where 1 class out of 21 is contributing 74% of the dataset. Also, the classes which are more in number in original dataset, has more words per sentence in them. If a sentence has more words, then our technique of data augmentation is likely to generate more sentences for it. After scaling factor of 4, there are few classes like "atis_cheapest" in dataset for which we are not able to generate more sentences. After scaling factor of 5, the rate of increase of data per class is reducing for classes with less samples in original data at much faster rate than scales which already dominate in terms of initial count of samples. By the end of the scaling with max possible data, the dominating class "atis_flight" is now occupying 85% of the data, which is even more than the original one. So, data is getting more unbalanced starting from scaling of 5.

(b) SNIPS: The graphs of 3 accuracy measures on SNIPS, i.e graph 4.8 , graph 4.9 and graph 4.10, shows a general trend of increase in accuracy with scaling. There is a small decrease in accuracy after the scale of 50 because gap increases in two classes "SearchCreativeWork" and "SearchScreaningEvent". In table 4.16 the classes "SearchCreativeWork" and "SearchScreaningEvent" are roughly equal in number of samples on scaling factor 50. Table 4.17 shows that the classes "SearchCreativeWork" and "SearchScreaningEvent" are not equal after scaling of more than 50. These two classes are closely related in terms of topic and word sharing. When we observe misclassifications on model, trained using only original data in table 4.9, we observe most of the misclassifications are between these two classes.

(c) **Analysis on $\mathbf{X} + \mathbf{X}_{dissimilar}$**:

class	% of contribution
$atis\_cheapest$	0.02
$atis\_ground\_service\#atis\_ground\_fare$	0.02
$atis\_aircraft\#atis\_flight\#atis\_flight\_no$	0.02
$atis\_airline\#atis\_flight_no$	0.05
$atis\_restriction$	0.1
$atis\_meal$	0.13
$atis\_capacity$	0.22
$atis\_distance$	0.28
$atis\_flight\_no$	0.30
$atis\_ground\_fare$	0.31
$atis\_city$	0.34
$atis\_airport$	0.42
$atis\_flight\#atis\_airfare$	0.45
$atis\_flight\_time$	0.87
$atis\_quantity$	0.90
$atis\_aircraft$	1.38
$atis\_abbreviation$	2.17
$atis\_airline$	3.46
$atis\_ground\_service$	5.72
$atis\_airfare$	8.50
$atis\_flight$	74.29

Table 4.10: ATIS Distribution : Scale 4

class	% of contribution
$atis\_cheapest$	0.02
$atis\_ground\_service\#atis\_ground\_fare$	0.03
$atis\_aircraft\#atis\_flight\#atis\_flight\_no$	0.03
$atis\_airline\#atis\_flight_no$	0.05
$atis\_restriction$	0.09
$atis\_meal$	0.13
$atis\_capacity$	0.22
$atis\_distance$	0.28
$atis\_flight\_no$	0.32
$atis\_ground\_fare$	0.32
$atis\_city$	0.35
$atis\_airport$	0.41
$atis\_flight\#atis\_airfare$	0.43
$atis\_flight\_time$	0.86
$atis\_quantity$	0.909
$atis\_aircraft$	1.37
$atis\_abbreviation$	2.06
$atis\_airline$	3.06
$atis\_ground\_service$	6.02
$atis\_airfare$	8.36
$atis\_flight$	74.07

Table 4.11: ATIS Distribution : Scale 5

class	% of contribution
$atis\_cheapest$	0.02
$atis\_ground\_service\#atis\_ground\_fare$	0.03
$atis\_aircraft\#atis\_flight\#atis\_flight\_no$	0.03
$atis\_airline\#atis\_flight_no$	0.05
$atis\_restriction$	0.09
$atis\_meal$	0.14
$atis\_capacity$	0.23
$atis\_distance$	0.27
$atis\_flight\_no$	0.34
$atis\_ground\_fare$	0.33
$atis\_city$	0.36
$atis\_airport$	0.40
$atis\_flight\#atis\_airfare$	0.43
$atis\_flight\_time$	0.86
$atis\_quantity$	0.92
$atis\_aircraft$	1.37
$atis\_abbreviation$	2.00
$atis\_airline$	3.63
$atis\_ground\_service$	6.29
$atis\_airfare$	8.24
$atis\_flight$	73.95

Table 4.12: ATIS Distribution : Scale 6

class	% of contribution
$atis\_cheapest$	0.002
$atis\_ground\_service\#atis\_ground\_fare$	0.02
$atis\_aircraft\#atis\_flight\#atis\_flight\_no$	0.005
$atis\_airline\#atis\_flight_no$	0.11
$atis\_restriction$	0.01
$atis\_meal$	0.02
$atis\_capacity$	0.05
$atis\_distance$	0.04
$atis\_flight\_no$	0.37
$atis\_ground\_fare$	0.06
$atis\_city$	0.09
$atis\_airport$	0.14
$atis\_flight\#atis\_airfare$	0.32
$atis\_flight\_time$	0.22
$atis\_quantity$	0.25
$atis\_aircraft$	3.61
$atis\_abbreviation$	0.46
$atis\_airline$	1.17
$atis\_ground\_service$	5.06
$atis\_airfare$	2.90
$atis\_flight$	85.07

Table 4.13: ATIS Distribution: Scale Max

class	% of contribution
RateBook	7.20
SearchScreeningEvent	11.38
SearchCreativeWork	13.66
PlayMusic	14.13
GetWeather	14.70
BookRestaurant	16.74
AddToPlaylist	22.18

Table 4.14: SNIPS Distribution : Scale 8

class	% of contribution
RateBook	5.74
SearchScreeningEvent	10.288
SearchCreativeWork	11.15
PlayMusic	11.56
GetWeather	14.17
BookRestaurant	15.66
AddToPlaylist	31.42

Table 4.15: SNIPS Distribution: Scale 25

class	% of contribution
RateBook	5.31
SearchScreeningEvent	10.48
SearchCreativeWork	10.33
PlayMusic	10.98
GetWeather	14.56
BookRestaurant	15.64
AddToPlaylist	32.67

Table 4.16: SNIPS Distribution: Scale 50

class	% of contribution
RateBook	4.49
SearchScreeningEvent	16.72
SearchCreativeWork	8.71
PlayMusic	9.46
GetWeather	13.97
BookRestaurant	16.21
AddToPlaylist	30.41

Table 4.17: SNIPS Distribution: Scale Max

i. ATIS: In table 4.18 of misclassifications on ATIS using X + $X_{dissimilar}$, we observe there are total 46 misclassifications.

   A. This model is able to reduce number of misclassfications compare to model trained using only original data.

   B. There are 5 samples numbered 94, 528, 654, 691 and 692 which are classified as "do not know class" or fallback class.

   C. This model is misclassifying 10 multiple-intent samples instead of 14 samples, which were misclassified when trained using only original data.

   D. There were 5 samples numbered 34, 36, 229, 492 and 499 which were misclassified because their classes are not present in training set (using only original data). Still, this is misclassifying them as we are not learning class name of unseen classes.

No	Actual Class	Predicted Class	Testing Utterance
12	atis_flight #atis_airfare	atis_flight	show flight and prices kansas city to chicago on next wednesday arriving in chicago by 7 pm
34	atis_day_name	atis_flight_time	what day of the week do flights from nashville to tacoma fly on
36	atis_day_name	atis_flight	what days of the week do flights from san jose to nashville fly on
50	atis_meal	atis_flight	what meals are served on american flight 811 from tampa to milwaukee
51	atis_meal	atis_flight	what meals are served on american flight 665 673 from milwaukee to seattle

94	atis_flight	fallback	does the airport at burbank have a flight that comes in from kansas city
108	atis_city	atis_flight	to what cities from boston does america west fly first class
114	atis_flight	atis_aircraft	show me the connecting flights between boston and denver and the types of aircraft used
138	atis_ground _fare	atis_ground _service	what are the fares for ground transportation in denver
164	atis_quantity	atis_capacity	how many canadian airlines international flights use aircraft 320
165	atis_quantity	atis_capacity	how many canadian airlines flights use aircraft dh8
176	atis_meal	atis_flight	what meals are available on dl 468 which al arrives in san francisco at 950 am
213	atis_flight #atis_airfare	atis_flight	list all sunday flights from cleveland to nashville and their fares
216	atis_distance	atis_airfare	how long does a flight from baltimore to san francisco take
229	atis_airfare #atis_flight	atis_airfare	list the airfare for american airlines flight 19 from jfk to lax
329	atis_meal	atis_flight	what meals are there on flight 382 from milwaukee to washington dc on tuesday morning
405	atis_flight #atis_airfare	atis_flight	i need a round trip flight from san diego to washington dc and the fares

406	atis_flight #atis_airfare	atis_flight	i need a round trip from atlanta to washington dc and the fares leaving in the morning
407	atis_flight #atis_airfare	atis_flight	i need a round trip from phoenix to washington dc and the fare leaving in the morning
469	atis_airfare	atis_flight	get the saturday fare from washington to toronto
492	atis_flight #atis_airline	atis_flight	i need flight and airline information for a flight from denver to salt lake city on monday departing after 5 pm
498	atis_flight #atis_airfare	atis_flight	i need flight and fare information departing from oakland to salt lake city on thursday before 8 am
499	atis_flight_no #atis_airline	atis_flight_no	i need flight numbers and airlines for flights departing from oakland to salt lake city on thursday departing before 8 am
501	atis_airport	atis_ground_fare	list airports in arizona nevada and california please
502	atis_airport	atis_ground_fare	list california nevada arizona airports
528	atis_flight_no	fallback	i need the flight numbers of flights leaving from cleveland and arriving at dallas
604	atis_flight #atis_airfare	atis_flight	show me the nonstop flights and fares from toronto to st. petersburg
621	atis_distance	atis_quantity	list distance from airports to downtown in new york
629	atis_city	atis_flight	list la

630	atis_city	atis_flight	list la
645	atis_airfare	atis_flight	list airfares for first class round trip from detroit to st. petersburg
654	atis_capacity	fallback	list seating capacities of delta flights from seattle to salt lake city
663	atis_flight	atis_flight #atis_airfare	give me the flights and fares for a trip to cleveland from miami on wednesday
688	atis_flight_no	atis_airline #atis_flight_no	flight number from dallas to houston
689	atis_flight_no	atis_quantity	flight number from houston to dallas
691	atis_flight_no	fallback	flight numbers on american airlines from phoenix to milwaukee
692	atis_flight_no	fallback	flight numbers from chicago to seattle
722	atis_flight	atis_quantity	how many northwest flights leave st. paul
723	atis_flight	atis_quantity	how many northwest flights leave washington dc
724	atis_flight	atis_quantity	how many flights does northwest have leaving dulles
735	atis_airport	atis_city	show me the airports serviced by tower air
739	atis_meal	atis_city	are meals ever served on tower air
740	atis_meal	atis_flight	are snacks served on tower air
773	atis_capacity	atis_quantity	how many people will a 757 hold
801	atis_flight	atis_quantity	how many flights does alaska airlines have to burbank

887	atis_airport	atis_ground_fare	tell me all the airports in the new york city area

Table 4.18: ATIS Misclassifications using X + $X_{dissimilar}$

ii. SNIPS: In table 4.19 on page 67 of misclassifications on SNIPS using X + $X_{dissimilar}$, we observe that there are total 19 misclassifications.

A. One of the missclassification belongs to "do not know class" or fallback category.

B. Most of the error sentences in this model maps with the model trained using only original data(X). This shows that this model is not adding any new type of error.

No	Actual Class	Predicted Class	Testing Utterance
13	SearchScreeningEvent	SearchCreativeWork	find on dress parade
16	GetWeather	SearchScreeningEvent	will the wind die down at my current location by supper time
27	SearchScreeningEvent	RateBook	in one hour find king of hearts
51	SearchCreativeWork	PlayMusic	play the new noise theology e p
90	SearchScreeningEvent	SearchCreativeWork	find now and forever
195	SearchCreativeWork	SearchScreeningEvent	where can i see the movie across the line: the exodus of charlie wright
345	SearchScreeningEvent	SearchCreativeWork	need to see mother joan of the angels in one second

408	SearchScreeningEvent	SearchCreativeWork	i had like to watch take this waltz
415	SearchCreativeWork	PlayMusic	play the song memories are my only witness
426	PlayMusic	SearchCreativeWork	play the song shine a light
429	SearchCreativeWork	PlayMusic	i want to listen to the song only the greatest
517	SearchCreativeWork	PlayMusic	listen to dragon ball: music collection
538	PlayMusic	SearchCreativeWork	do you have something like impossible is nothing by abderrahmane abdelli
565	SearchCreativeWork	fallback	can you find me the back when i knew it all album
579	SearchCreativeWork	SearchScreeningEvent	show me heavenly sword
592	SearchScreeningEvent	SearchCreativeWork	i would like to book a highly rated brasserie with souvlaki neighboring la next week
618	SearchCreativeWork	PlayMusic	play the album journeyman
654	AddToPlaylist	PlayMusic	i want to add a song by jazz brasileiro
664	SearchScreeningEvent	SearchCreativeWork	where can i see a slice of life

Table 4.19: SNIPS Misclassifications using X + $X_{dissimilar}$

(d) **Analysis on X + $\mathbf{X}_{similar}$ + $\mathbf{X}_{dissimilar}$**:

i. ATIS: In table 4.20 of misclassifications on ATIS using X + $X_{similar}$+ $X_{dissimilar}$, we observe there are total 40 misclassifications.

   A. This model is able to reduce the number of misclassifications.

   B. There are 7 samples numbered 114, 202, 270, 271, 406, 787 and 888 which model is classifying as "Do not know class" or fallback class.

   C. We observe model misclassifying 11 multiple-intents samples instead of 14 in original. Including learning "fallback intent" on 2 of them.

   D. Error sentences mostly overlap with original error sentences. This shows that model is not adding any new type of error.

   E. There were 5 samples numbered 34, 36, 229, 492 and 499 which were misclassified because their classes are not present in training set (using only the original data). Still, this model is misclassifying them because it is not learning class name of unseen classes.

No	Actual Class	Predicted Class	Testing Utterance
12	atis_flight #atis_airfare	atis_flight	show flight and prices kansas city to chicago on next wednesday arriving in chicago by 7 pm
34	atis_day_name	atis_flight	what day of the week do flights from nashville to tacoma fly on
36	atis_day_name	atis_flight	what days of the week do flights from san jose to nashville fly on
94	atis_flight	atis_airport	does the airport at burbank have a flight that comes in from kansas city

114	atis_flight	fallback	show me the connecting flights between boston and denver and the types of aircraft used
137	atis_ground_fare	atis_airfare	what's the fare for a taxi to denver
168	atis_airport	atis_airline	which airport is closest to ontario california
176	atis_meal	atis_flight	what meals are available on dl 468 which al arrives in san francisco at 950 am
202	atis_aircraft	fallback	at the charlotte airport how many different types of aircraft are there for us air
206	atis_flight #atis_airfare	atis_airfare	list all flights and their fares from indianapolis to memphis on a monday morning
213	atis_flight #atis_airfare	atis_flight	list all sunday flights from cleveland to nashville and their fares
229	atis_airfare #atis_flight	atis_airfare	list the airfare for american airlines flight 19 from jfk to lax
243	atis_capacity	atis_flight_no	what is the seating capacity for delta be1
270	atis_flight	fallback	cleveland to kansas city arrive monday before 3 pm
271	atis_flight	fallback	kansas city to cleveland flight arrive wednesday before 5 pm
405	atis_flight #atis_airfare	atis_flight	i need a round trip flight from san diego to washington dc and the fares
406	atis_flight #atis_airfare	fallback	i need a round trip from atlanta to washington dc and the fares leaving in the morning

407	atis_flight #atis_airfare	atis_airfare	i need a round trip from phoenix to washington dc and the fare leaving in the morning
492	atis_flight #atis_airline	atis_flight	i need flight and airline information for a flight from denver to salt lake city on monday departing after 5 pm
498	atis_flight #atis_airfare	atis_flight	i need flight and fare information departing from oakland to salt lake city on thursday before 8 am
499	atis_flight_no #atis_airline	atis_flight_no	i need flight numbers and airlines for flights departing from oakland to salt lake city on thursday departing before 8 am
502	atis_airport	atis_flight	list california nevada arizona airports
570	atis_flight	atis_abbreviation	baltimore to kansas city economy
604	atis_flight #atis_airfare	atis_airfare	show me the nonstop flights and fares from toronto to st. petersburg
629	atis_city	atis_flight	list la
630	atis_city	atis_flight	list la
654	atis_capacity	atis_flight	list seating capacities of delta flights from seattle to salt lake city
663	atis_flight	atis_flight #atis_airfare	give me the flights and fares for a trip to cleveland from miami on wednesday
722	atis_flight	atis_quantity	how many northwest flights leave st. paul
723	atis_flight	atis_quantity	how many northwest flights leave washington dc

724	atis_flight	atis_quantity	how many flights does northwest have leaving dulles
735	atis_airport	atis_city	show me the airports serviced by tower air
740	atis_meal	atis_flight	are snacks served on tower air
773	atis_capacity	atis_quantity	how many people will a 757 hold
779	atis_aircraft	atis_abbreviation	tell me about the m80 aircraft
780	atis_aircraft	atis_abbreviation	tell me about the m80 aircraft
787	atis_flight	fallback	list all flights on all types of aircraft arriving in denver between 8 and 9 pm
801	atis_flight	atis_quaw	many flights does alaska airlines have to burbank
881	atis_airline	atis_flight	is there one airline that flies from burbank to milwaukee milwaukee to st. louis and from st. louis to burbank
888	atis_flight	fallback	please find all the flights from cincinnati to any airport in the new york city area that arrive next saturday before 6 pm

Table 4.20: ATIS Misclassifications using X + $X_{similar}$ + $X_{dissimilar}$

ii. SNIPS: In table 4.21 of misclassifications on SNIPS using X + $X_{similar}$+ $X_{dissimilar}$, we observe there are total 22 misclassifications.

A. This model is classifying 3 out of the 22 misclassifications as "do not know class" or fallback class.

B. The number of misclassifications are same when compared to model trained using only original data. Though, 3 of them are now clas-

sified as fallback.

No	Actual Class	Predicted Class	Testing Utterance
16	GetWeather	SearchScreeningEvent	will the wind die down at my current location by supper time
27	SearchScreeningEvent	SearchCreativeWork	in one hour find king of hearts
51	SearchCreativeWork	PlayMusic	play the new noise theology e p
90	SearchScreeningEvent	SearchCreativeWork	find now and forever
103	SearchScreeningEvent	SearchCreativeWork	can i get the butterfly crush showings
156	GetWeather	fallback	at meal time while i m here will it be hot
266	GetWeather	BookRestaurant	i need a table in uruguay in 213 days when it s chillier
326	SearchCreativeWork	PlayMusic	play the electrochemical and solid state letters song
345	SearchScreeningEvent	SearchCreativeWork	need to see mother joan of the angels in one second
361	SearchScreeningEvent	SearchCreativeWork	i want to go see the trouble with girls
434	RateBook	SearchCreativeWork	the book history by contract is rated five stars in my opinion
517	SearchCreativeWork	PlayMusic	listen to dragon ball: music collection
529	RateBook	SearchScreeningEvent	put four rating on the raging quiet

538	PlayMusic	fallback	do you have something like impossible is nothing by abderrahmane abdelli
542	SearchCreativeWork	SearchScreeningEvent	where can i find thor meets captain america
579	SearchCreativeWork	SearchScreeningEvent	show me heavenly sword
587	SearchCreativeWork	SearchScreeningEvent	find a man needs a maid
592	SearchScreeningEvent	SearchCreativeWork	find the panic in needle park
614	SearchCreativeWork	SearchScreeningEvent	i'm looking for circus world
654	AddToPlaylist	PlayMusic	i want to add a song by jazz brasileiro
660	GetWeather	fallback	show weather while sunset in the same area in south carolina
664	SearchScreeningEvent	SearchCreativeWork	where can i see a slice of life
697	SearchCreativeWork	SearchScreeningEvent	find politicsnation with al sharpton

Table 4.21: SNIPS Misclassifications using X + $X_{similar}$ + $X_{dissimilar}$

4.4 Conclusion

In this paper, we proposed a technique to augment similarity and dissimilarity data for intent classification and slot filling in Spoken Language Understanding systems of a virtual digital assistant.

1. Using similarity data, we have shown that our data augmentation technique has helped in outperforming the state of the art systems. Our "X+ $X_{similarity}$"

model is doing better than the previous state of the art systems in all the three accuracy measures. In ATIS, we have increased the intent accuracy by 2% (see graph 4.3). In Snips, we have increased the intent accuracy by 0.44% (see graph 4.8).

2. We have proposed a new way to calculate intent accuracy which can incorporate the risk of misclassification in SLU tasks. Our experiments using dissimilar sentences have shown that we can train a system which is more immune to misclassification. In error analysis on ATIS using "X + $X_{dissimilar}$" data, we have found that the number of misclassifications has reduced. Also, we are able to classify 3 test samples as "do not know category". These 3 samples were misclassified when we had used only original data for learning.

3. We have also shown that using a joint model with similar and dissimilar meaning sentences, we can generalize better. In error analysis on ATIS using "X + $X_{similar}$ + $X_{dissimilar}$" data, we have decreased the total number of misclassifications. The number of misclassifications is lesser than both the models, one using only X and other using X + $X_{dissimilar}$. This shows that we were able to create a more robust system.

4.5 Future Work

In this work, we observed that the current systems, including this work, are able to handle single intent utterances well. But they do not efficiently handle the utterances with multiple intents and their order. In future, it will be important to address the issue of multiple intents and their order in an utterance.

The embeddings that we are using in this work are not dynamic and do not take into account word sense to give a vector representation of a word. We used static word

embeddings which will give same representation of a word irrespective of its sense. There are new dynamic word embeddings such as "ELMo" (Peters *et al.* (2018)) and "BERT" (Devlin *et al.* (2018)) which give vector representation of a word based on its context in a given sentence. It will be useful to observe the impact of using dynamic word embeddings as part of the future work.

Furthermore, The publicly available datasets for intent prediction and slot-filling do not take into account out of domain instances. There is a need to create a dataset which contains adversarial intents, negation intents and completely out of domain scenarios to test the robustness of the system.

REFERENCES

Banerjee, S. and T. Pedersen, "An adapted lesk algorithm for word sense disambiguation using wordnet", in "International conference on intelligent text processing and computational linguistics", pp. 136–145 (Springer, 2002).

Basile, P., A. Caputo and G. Semeraro, "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 (2014).

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

Goo, C.-W., G. Gao, Y.-K. Hsu, C.-L. Huo, T.-C. Chen, K.-W. Hsu and Y.-N. Chen, "Slot-gated modeling for joint slot filling and intent prediction", in "Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)", vol. 2, pp. 753–757 (2018).

Guo, D., G. Tur, W.-t. Yih and G. Zweig, "Joint semantic utterance classification and slot filling with recursive neural networks", in "Spoken Language Technology Workshop (SLT), 2014 IEEE", pp. 554–559 (IEEE, 2014).

Haffner, P., G. Tur and J. H. Wright, "Optimizing svms for complex call classification", in "Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on", vol. 1, pp. I–I (IEEE, 2003).

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

Kurata, G., B. Xiang, B. Zhou and M. Yu, "Leveraging sentence-level information with encoder lstm for semantic slot filling", arXiv preprint arXiv:1601.01530 (2016).

Lee, Y. K., H. T. Ng and T. K. Chia, "Supervised word sense disambiguation with support vector machines and multiple knowledge sources", in "Proceedings of SENSEVAL-3, the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text", (2004).

Lesk, M., "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 (ACM, 1986).

Liu, B. and I. Lane, "Attention-based recurrent neural network models for joint intent detection and slot filling", arXiv preprint arXiv:1609.01454 (2016).

Mesnil, G., Y. Dauphin, K. Yao, Y. Bengio, L. Deng, D. Hakkani-Tur, X. He, L. Heck, G. Tur, D. Yu *et al.*, "Using recurrent neural networks for slot filling in spoken language understanding", IEEE/ACM Transactions on Audio, Speech, and Language Processing **23**, 3, 530–539 (2015).

Peng, B. and K. Yao, "Recurrent neural networks with external memory for language understanding", arXiv preprint arXiv:1506.00195 (2015).

Peters, M. E., M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee and L. Zettlemoyer, "Deep contextualized word representations", arXiv preprint arXiv:1802.05365 (2018).

Raymond, C. and G. Riccardi, "Generative and discriminative algorithms for spoken language understanding", in "Eighth Annual Conference of the International Speech Communication Association", (2007).

Sutskever, I., O. Vinyals and Q. V. Le, "Sequence to sequence learning with neural networks", in "Advances in neural information processing systems", pp. 3104–3112 (2014).

Tan, L., "Pywsd: Python implementations of word sense disambiguation (wsd) technologies [software]", https://github.com/alvations/pywsd (2014).

Tur, G., D. Hakkani-Tür and L. Heck, "What is left to be understood in atis?", in "Spoken Language Technology Workshop (SLT), 2010 IEEE", pp. 19–24 (IEEE, 2010).

Xu, P. and R. Sarikaya, "Convolutional neural network based triangular crf for joint intent detection and slot filling", in "Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on", pp. 78–83 (IEEE, 2013).

Zhong, Z. and H. T. Ng, "It makes sense: A wide-coverage word sense disambiguation system for free text", in "Proceedings of the ACL 2010 system demonstrations", pp. 78–83 (Association for Computational Linguistics, 2010).

APPENDIX A

EXPERIMENTS ON INDUSTRIAL SYSTEMS

A.1 Initial Setup

The experiment is performed on three different industrial voice assistants:

1. Google DialogFlow [1]
2. IBM Watson Assitant [2]
3. Microsoft Luis [3]

We choose above three systems for our experiment. In the demo, we are showing that the above mentioned system fails on various user utterances. On some user utterances, these system shows very less confidence where higher confidence was expected, or, have shown higher confidence where lower confidence or no confidence was expected.

First, we checked for a similar meaning sentence, using similar words but not same words. For this test, we have created an intent name *"Test"* in all the three virtual voice assistants, trained using user utterance *"I like the way you see her"*. The process used to train is shown in the demo video [4]

The figure A.2 on page 80 shows that there is a significant drop, when tested on sentence *"I love the way you view him"* in the similar meaning sentences when similar words are used instead of actual one. In the 2^{nd} experiment, we are using the same trained model as used above and running it against the input *"I do not like the way you see her"*. We see a higher confidence in prediction where the contextual meaning of the intent is not same. you can see the results in figure A.1 on page 80.

The above two experiment shows that there is a possibility of False Positives and False Negative in current industrial systems.

[1] `https://dialogflow.com`

[2] `https://www.ibm.com/watson/ai-assistant`

[3] `https://www.luis.ai/home`

[4] `https://www.youtube.com/watch?v=dZcc5AZzoKM&feature=youtu.be`

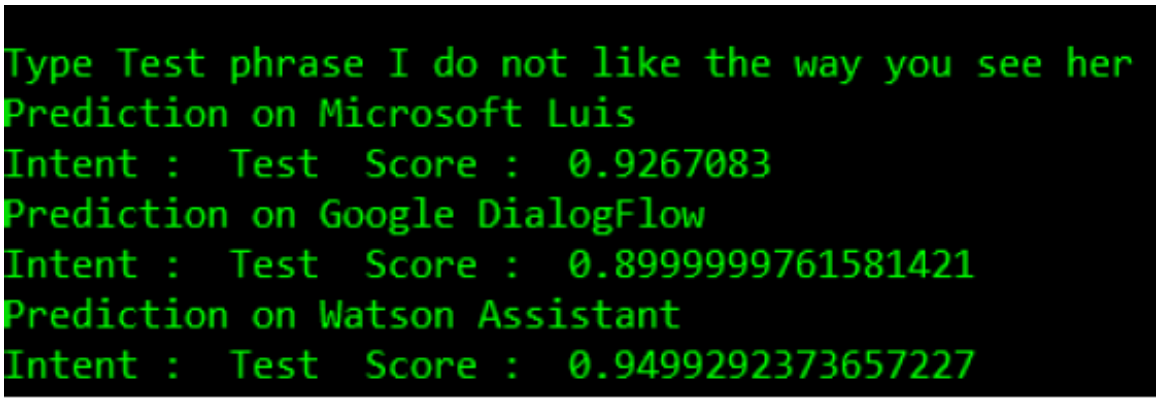


Figure A.1: Results for Adversarial Example

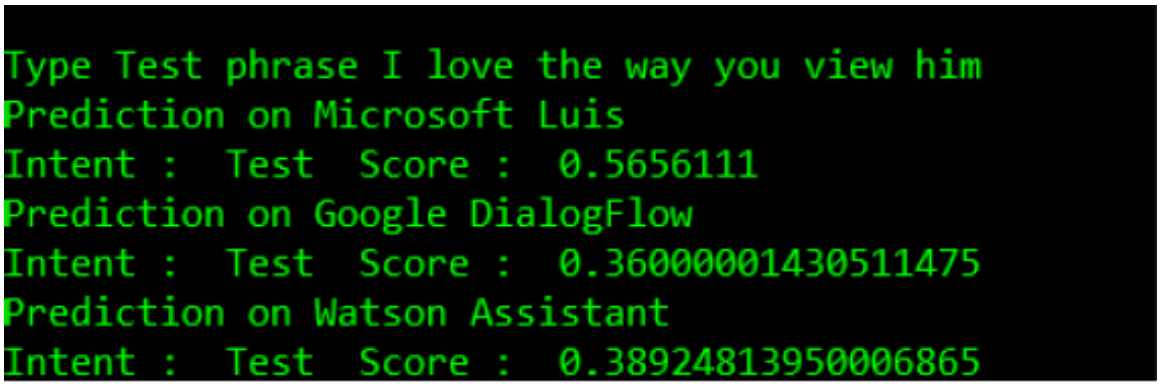


Figure A.2: Results for Similar Example

P.S. These results are posted using industrial system in May 2018. The results may change if tested on current system.

A.2 Proposed Method

A.2.1 Overview

The industrial systems for intent classification requires a user to cover possible scenarios for intent classification. This task is hard and makes the natural language processing tool usability limited to how many use case a user can cover while defining the training phrases for an intent. The purpose of the NLU system is to minimize this

effort of the user. The first idea here is how we can automatically extend the training phrases without the requirement of user training phrases. For Example, if a user gives a training phrase *"Do the dance"*, we can automatically using prior knowledge and NLP tools like nltk, create similar sentences of this sentence like [*"Perform the dance"*, *"Do the dancing"*, *"Perform the dancing"*].

Then another problem we encounter is that these models fail to recognize the negation of a sentence. Given a sentence *"I like to see you dance"* and a sentence *"I hate to see you dance"* should not be classified in same intent. The second idea is to incorporate this negation knowledge into the algorithm having a negative impact on the confidence level for an intent classification.

The third problem we see is to add knowledge of differentiation of a sentence based on whether it is a question or a statement. If an intent *"I"* is about questions related to the topic *"T"*, then a statement which is providing knowledge about the topic ***T*** but is not asking about the knowledge related to Topic *"T"*, should not be recognized as intent *"I"*. The current systems show drops in confidence in some cases, but the drop is not significant enough to classify it correctly. All these scenarios mentioned above are considered while designing the current algorithm.

A.2.2 Adding Similar Sentences Score

To incorporate the similar sentence meaning into the prediction. We can train a Model $M_{similarity}$ which takes similar sentences extracted using nltk from the training phrases. The automatic process involves using the part-of-speech tagging tool to replace verbs, adverbs, nouns, adjectives, etc. using wordnet lemmas relation. Then use these sentences to train a different model which has more training phrases and tries to learn same class (Intent). Let us suppose $P(Y|X, M)$ represents the probability of intent Y, given the utterance by the user a sentence X and using model M (this

model M represents the current model used in virtual assistant systems). Then we will calculate the probability of P(Y|X,$M_{similarity}$), now we are predicting the probability of intent Y, given the utterance by the user a Sentence X and using Model $M_{similarity}$. A general representation of the relation can be

$$Score = \frac{W_1 * P(Y|X,M) + W_2 * P(Y|X,M_{similarity})}{W_1 + W_2} \tag{A.1}$$

Where W_1 and W_2 represent the weight of respective probabilities. We can say that current System is a specific form of this where W2 is zero. It will help reduce the False Negatives currently present in the system.

A.2.3 Checking Negation Semantic

The current systems do not consider the natural language contextual meaning into account, specifically in cases where words used in user utterances are same to that of a training phrase but the contextual meaning of training intent and a user is completely opposite. This time we can train a model $M_{negation}$, this model takes into account negation of the verbs defined in the training phrases. We will use part of speech tagging and wordnet lemmas to find antonyms and create sentences which are opposite in meaning to the current intent. We names it as P($Y_{negation}$| X, $M_{negation}$) represents the probability of intent $Y_{negation}$, given the user utterance is X and using the model $M_{negation}$. Consider a function F(x,y), given as:

$$F(x,y) \colon x \leftarrow x >= y$$

$$0 \leftarrow otherwise$$

Now to find the score of an intent we check for negation using this:

$$Score = \frac{W_1 * P(Y|X,M) + W_2 * P(Y|X,M_{similarity})}{W_1 + W_2} \tag{A.2}$$

$$Score_{new} = F(Score, P(Y_{negation}|X, M_{negation}) \tag{A.3}$$

This takes care of classification into sentences which are more inclined towards the opposite semantic meaning. This way the new Score prediction takes care of the False positives and makes the system more robust.

A.3 Results

1. Result on Similarity : On running the *"I like the way you view her"*, on the same trained model we have in section 4. Result can be seen in figure A.3 on page 84. We see improvement in accuracy results, as per expectation.

2. On running the *"I dislike the way you see her"*, on the same trained model we have in section 4. Result can be seen in figure A.4 on page 84. We see decrease in accuracy result, as per expectation.

.

A.4 Conclusion

Currently, after every few days we hear a news about a voice agent doing something which it was not intended to do. This paper focuses on reducing this issue by making intent classifier more robust using prior knowledge. Experiments have shown that current system can be improved using prior knowledge of similarity and dissimilarity of words and sentences.

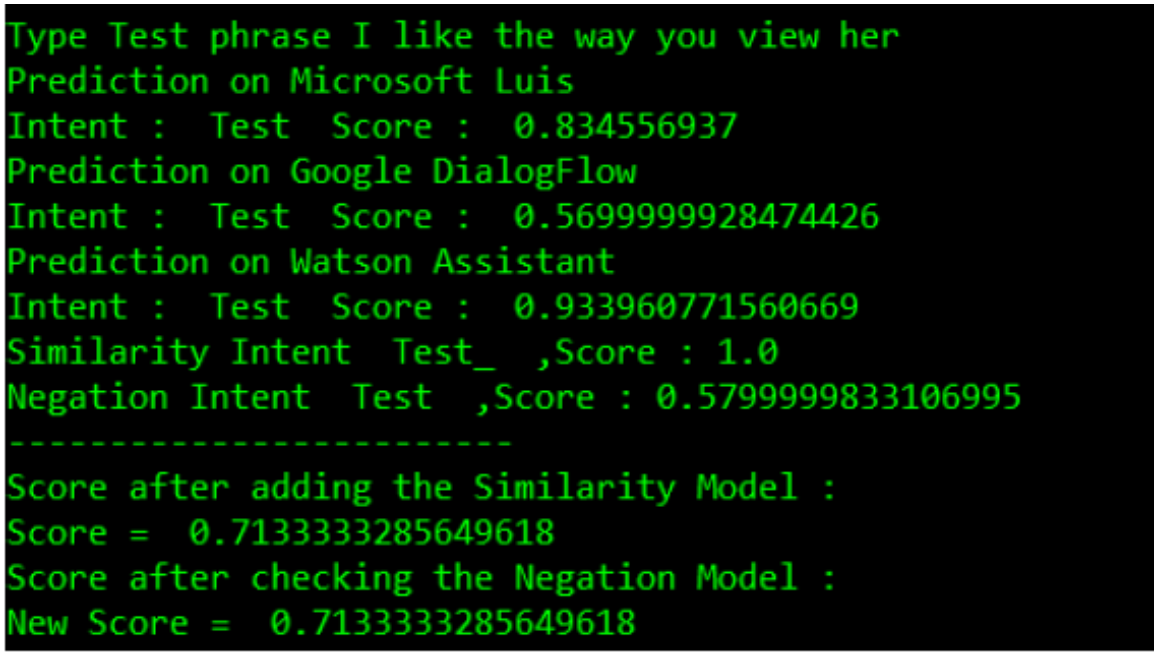


Figure A.3: Result on Proposed Similarity Model

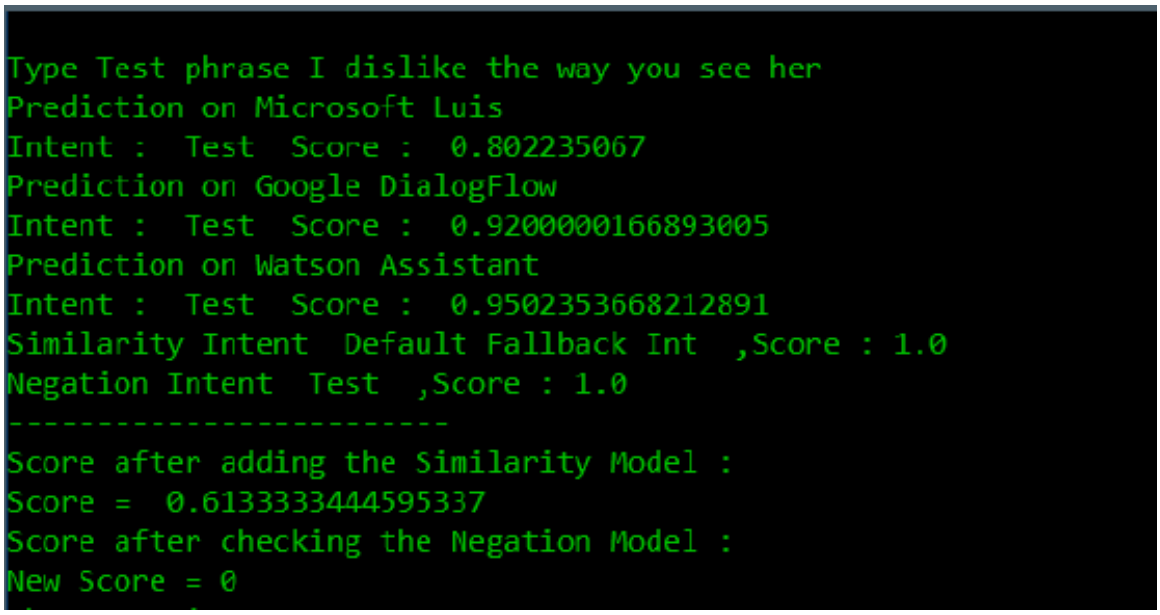


Figure A.4: Result on Proposed Dissimilarity Model