Preface

This report is part of Natural Language Processing Course offered by Prof. Sutanu Chakraborti.

November 2015

Yogesh B
EE12b066
IIT Madras

Table of Contents

Natural Language Processing

Chapter 1

Yogesh Balaji

IIT Madras, Chennai, India
ee12b066@ee.iitm.ac.in

Abstract. Natural Language Processing (NLP) is a field of Artificial Intelligence that is concerned with understanding human language. The subject is developed over time with different people looking at it with different perspectives. This chapter addresses some foundational issues in NLP and motivates why problems in NLP are hard. Some standard problems in NLP are introduced challenges associated with the problems are pointed out. The final section of the chapter also talks about the empirical methods in NLP and how effective they they have become in the recent years. ...

Keywords: Natural Language Processing, Artificial Intelligence

1 Introduction

Natural language processing is a branch of artificial intelligence and computational linguistics that deals with analyzing, understanding and generating the languages that humans use naturally in order to interact with the external world. The field started off in 1950's with gradual improvement through the 80's when people started building simple models for applications like machine translation. This period was influenced by people like Noam Chomsky who proposed significant ideas on linguistics. There was a dramatic change through 80's when people started using machine learning techniques for NLP. This was facilitated by steady increase in computational power which made it possible to run algorithms on huge datasets.

Problems in NLP are known to be hard. This is because of the fact that we don't consciously understand language ourselves. We keep using language without consciously being aware of how we are using them. Another reason why NLP is hard is because of ambiguity. Many sentence structures tend to imply the same meaning and so deciding which one to choose becomes hard. Much of the NLP tasks revolves around breaking ambiguity at multiple levels.

2 Overview of Natural Language Processing

2.1 Classes of NLP

NLP systems can be broadly classified into two categories - Natural Language Understanding(NLU) and Natural Language Generation. Natural language understanding deals with machine reading comprehension (i.e) the goal of NLU is

to enable a system interpret an input text fragment. The process of interpretation can be viewed as a translation of the text from natural language to a representation in an unambiguous formal language. On the other hand, Natural language understanding represents the task of generating natural language from a machine representation system such as a knowledge base or a logical form. Both these tasks can be seen to be complementary to each other.
Natural Language Understanding tasks are generally harder than Natural Language Generation. This is because of the specificity associated with NLU tasks. Due to the inherent ambiguity in the language, there can be many representations for a single piece of text. All of these representations are valid for a NLG system, but a NLU system needs to pick the right representation among all the many. So, there is more room for error in case of NLG as compared to NLU. We can think of this as flight taking off and langing. It is generally easier to take off a flight because after the take-off even if the flight drifts in any direction, it can be adjusted. But while landing, it becomes essential that we land at the right location. In our case, flight take-off is similar to NLG and landing is similar to NLU.

2.2 The task of understanding

Understanding Natural Language becomes a crucial component in most of the NLP tasks. But, what does it mean to understand something? One pragmatic view to this question is to think of what we can do as a result of understanding. In order to make machines understand like humans, we need to formalize our thought processes. But this becomes a hard task because of its recursive nature. The idea to formalize our thoughts is also another thought. This is similar to a problem in Machine Learning - the taks of choosing the best classifier from a given set of classifiers. This is a very hard problem because choosing the best classifier is another classification task.
The main difficulty in NLP is that we ourself don't understand completely how we use our language. We learn things without being aware of how we learn them. It is this unconscious component of knowledge that becomes very hard to formalize. We can think of human knowledge as an iceberg - the conscious component is the tip of iceberg above the sea surface, whereas the unconscious component is the iceberg under water.
People have tried to make good inferences using only this small component of conscious knowledge. One such system is the Amazon Recommender system that makes recommendation to the users based on their purchase history. Amazon's recommender system is based on a technique called "collaberative filtering". The basic idea in this method is that people having similar liking for an item tend to buy similar things. Effectively, people are categorized into bins based on their surface level footprints. For example, for recommending books, if an user has a liking for certain kinds of books, he might have a liking for other books which were liked by people with similar footprints.

3 Influential people in Natural Language Processing

Two important people in NLP were Noam Chomsky and Roger Schank whose ideas influenced the field through the late 20^{th} century. Chomsky was a linguist who was mostly concerned with patterns in language and language modelling. Chomsky has argued that linguistic structures are at least partly innate, and that they reflect a "universal grammar" that underlies and can account for all human grammatical systems. The Chomskyan approach towards syntax, often termed generative grammar, studies grammar as a body of knowledge possessed by language users. Chomsky strongly opposed statistical methods in language modelling. On the symposium "Brains, Minds and Machines", he pointed out that the "new AI" - focused on using statistical learning techniques to better mine and predict data is unlikely to yield general principles about the nature of intelligent beings or about cognition.

Roger Schank's theory, on the other hand was concerned more with the connection between language and memory. His claim was that given a piece of text, we don't process it word by word. Instead, we already have a model of the subject and the given piece of text just fills in some specific details pertaining to the model. For instance, if we read an article about accidents, we have a general notion what happens in an accident, so the given piece of text just fills in details like place, type of accident, death count, etc. Sometimes, there are some exceptions to the general trend observed and we take note of them. His work on language and memory was grounded in an elegant formalism, but this system becomes impractical as there are a huge pool of ideas. So, his work wasn't very famous.

4 Basic concepts in Natural Language Processing

Approaches in NLP can be divided broadly into four hierarchies: Character $->$ Word $->$ Sentence $->$ Discourse. Different tasks need approaches at different levels. It is important to see that the atomic unit of meaning is "word" as characters by themselves don't give any meaning. Sentences are a collection of words following a specific sequence, not just any random collection. We cannot simply reduce sentence into words and analyze them. We can correlate this with Chemistry, where electrons are letters, words are atoms and sentences are molecules. Any random atoms sequence of atoms cannot come together to form molecules, some specific atoms bond with others. Similarly, all electrons are alike in their properties, but the properties of atoms differ.

4.1 Properties of words

The four basic properties associated with words are morphology, lexical semantics, phonetics and parts of speech. Let us look at each one of these properties.

Morphology Morphology deals with the structure of words. For instance, it's about studying why 'a','p','p','l' and 'e' come together to form apple. The first question to address is why to study morphology at all. The main reason why we are instested in morphology is because there are many words derived from some base words. For example, singing inflects from "sing" and happiness is derived from "happy". So, an understanding of morphology would assign meanings to parts of words, which will help in reducing the analysis of huge space of words.

Lexical Semantics Lexical Semantics deals with the task of assigning meanings to individual words. Encoding meaning is a very hard task because we keep using words subconsciously without understanding how we learn each word. Lexical semantics looks at how the meaning of the lexical units correlate with the structure of language. The study of lexical semantics deals with the classification and decomposition of lexical terms, the differences and similarities in lexical semantic structures and the relationship of the lexical meaning to sentence meaning and syntax.

Phonetics Phonetics is the field of linguistics that focusses on speech. It is concerned with the physical properties of speech sounds and signs - their physiological production, acoustic properties, auditory perception, and neurophysiological status.

Parts of Speech A parts of speech is a category of words having similar grammatical properties. Words that are assigned to the same part of speech generally display similar behavior in terms of syntax - they play similar roles within the grammatical structure of sentence and sometimes in terms of morphology. Common English parts of speech are noun, verb, adjective, adverb, pronoun, preposition, conjunction and interjection.

4.2 Sentences

Sentences are collection of words which come together following a specific sequence. To analyze sentences, we study compositional semantics. Compositional semantics, unlike lexical semantics focusses on how small units combine to form the meaning of a larger unit. The meaning of a sentence is determined by the meaning of its words in conjunction with the way they are syntactically combined. However, there are exceptions to compositionality which comprise of anomalies, idioms, ambiguities and presuppositions.

4.3 Discourse

Discourse level processing deals with understanding the meaning of a set of sentences. Understanding discourse is very hard because of the complex nature of the task. One example of the discourse level task is "Anaphora Resolution".

Anaphora is the use of an expression the interpretation of which depends upon another expression in context. For example, given the sentences "Patel dropeed the plate. It shattered loudly", "it" is an anaphora that refers to the plate. Understanding what an anaphora implies is a hard task.

5 Machine Learning methods

In the past few years, machine learning have started to play a donimant role in Natural Language Processing systems. People have tried to use different statistical models for differennt tasks and have celebrated some success. However, this is not the end of it. We may train a model and get some parameters, but we are not sure if this is a sufficient model and the model would work in a generalized setting. But, the machine learning techniques have definitely helped in easing the problem of Knowledge Acqusition Bottleneck.Even as the machine learning system show good success, they still rely on humans. Computers can give the result, but it is the human who interprets it.

A dictionary or just a corpus can't model everything. In order to interpret certain words, it is necessary for the machine to do what humans do (for example, feel like the humans). Also, a machine learning system will not be able to model everything accurately. There is a quote that beautifully explains this - "All models are wrong, however, some are useful". Every model will have an inherent "inductive bias" that represents how complex the model is. For instance, a linear regression system can't model quadratic curve. This is the inductive bias associated with the linear system. In the similar sense, every model makes some inherent assumption about the data and so models are not fully accurate.

6 Classical Artificial Intelligence Systems

In the previous sections, we have made a lot of claims that NLP is a field of Artificial Intelligence. Let us look into this issue more deeply. When can we say that a problem is a AI problem? In an AI problem, unlike other standard problems, the sequence of steps is not specified. Only the final objective is mentioned, but the process of doing things is not stated. The algorithm should this process so as to reach the end goal. Searching is generally easier in AI problems and so any given problem is mapped to a search problem. One example is the declarative language PROLOG.

A classical AI problem is generally governed by rules and models. One example is the diagnosis system which uses Noisy Channel model. But there are some problems where rules and models are not given by humans. One example is the decision trees in which the rules are learnt. So, it is not a classical AI problem. As far as the Natural Language is concerned, we don't have well defined models and so NLP doesn't fall under classical AI systems.

7 Conclusion

In this capture, we looked at a basic introduction to the subject of Natural Language Processing. We discussed various problems in the field and motivated why NLP is hard. We then looked at the ideas of some influential people in the field. Finally, we discussed the machine learing methods that have been successful in the last few decades.

References

1. Class lecture notes
2. Wikipedia
3. Stanford NLP lecture notes

Chapter 2

Yogesh Balaji

Indian Institute of Technology, Madras
ee12b066@ee.iitm.ac.in

Abstract. The previous chapter addressed some foundational issues in NLP and it gave a basic introduction to the field. The difficulty of the problems and the views of some of the influential people were discussed. This chapter is essentially a critique of the traditional views reported in the previous chapter.

Keywords: Natural Language Processing, Artificial Intelligence, critique

1 The task of understanding

The fundamental issue behind understanding language is that we are not consciously aware of how we use the language. So, we rely only on the conscious knowledge to make our decisions. But, the recent upsurge of big data has enabled us to crowd-source the opinions of people following similar footprints, resulting in systems like the Amazon Recommender Systems. Even though these techniques of collaborative filtering achieves good performance to some extent, it is nowhere close to solving the fundamental issue. These methods can be seen as heuristic approximations to the main problem.

I feel its important to concentrate on the fundamental issue rather than these approximations. With dramatic rise in the storage and computational power, modelling unconscious knowledge could be achieved to some extent. For instance, all data acquired by the child as he/she grows could potentially be collected from which it might be possible to model the unconscious knowledge. In another line of thought, instead of just collecting the data, we might ask some questions to a child at different stages of like. From this, we can try to model the correlation in the thought process over age, and this could provide some insight on the unconscious knowledge.

2 Ideas of influential people revisited

As discussed in section 3 of Chapter 1, Chomsky and Schank were the two important people whose ideas influenced the field of NLP through the late 20^{th} century. Chomsky focussed on grammar based approach for language modelling and strongly opposed statistical measures claiming that statistical measures might

not capture general principles. I agree to the first part of the argument, syntactic approaches are essential for language modelling. But, I do not agree with a strong no to statistical measures. In the last few years, with the surge of machine learning methods, people have started moving completely to corpus based methods and so Chomskian theories started vanishing. This is not desirable as pure statistical methods will not solve all problems. Instead, I feel people have to resort to a mix of both and that can fetch good results.

Now, let us look at Roger Schanks theory. Roger Schank focussed more on connection between language and memory. He was strongly of the opinion that memory and experiences play a vital role in learning. People werent convinced by his ideas citing that it is highly impractical. But with increase in computational power, this idea can become feasible. As discussed previously, capturing all the data a child acquires and all actions the child performs if collected, could make this system feasible.

2.1 Machine Learning methods

Section 5 of Chapter 1 covers the advantages and shortcomings of machine learning approaches in NLP. Machine learning has been succesful in that it helped resolve the issue of Knowledge Acqusition Bottleneck. But, I feel that a naive corpus based learning doesnt capture much of information as we humans do. For instance, humans associate words with feelings, emotions, etc. that guides them significantly in the process of understanding a text. These attributes could not be captured just by looking at text, but needs a much more complex model. One way to better the system is incorporating memories and experiences which inherently have emotions in them. This is a direct consequence of Roger Schanks ideas. As the field of data science is growing enormously, it would become capable of handling complex models. We should leverage these benefits and try to include many other sources of knowledge into the models which could help capture complex dependencies in the language.

3 Conclusion

In this chapter, we presented a critique of various notions in the field of NLP. We first discussed the task of understanding. Then, we analysed the views of Chomsky and Schank. Finally, we looked at machine learning techniques and the potential improvements that could be incorporated.

References

Class lecture notes

Chapter 3

Yogesh Balaji

Indian Institute of Technology, Madras
ee12b066@ee.iitm.ac.in

Abstract. The previous two chapters discussed the foundational issues in NLP and motivated why the problems in NLP are hard. In this chapter, we shall look at some standard problems in NLP and discuss techniques used to solve them. We shall also provide some important ideas behind each of the topics and establish connections between them

Keywords: Natural Language Processing, Artificial Intelligence

1 Some classic problems in NLP

Problems in NLP typically fall in two categories - Natural Language Understanding, that focusses on systems that understand a given piece of text and Natural Language Generation, that encompasses systems that can generate text in Natural Language. There are many systems that involve a fusion of both these methods (ie) these tasks focus on making sense of a given piece of text and then using this knowledge to generate new text. One example of this is the task of Machine Translation where we need to understand a piece of text in one language and then generate the information in another language.

One of the simple problems in NLP that we could use as a starting point in our discussion is the problem that is ubiquitous and has enormous utility in our everyday lives - Spell Check. The task involves detecting errors in text and making relevant suggestions. After this, we shall discuss about Information Extraction, where we are interested to retrieve documents based on our query. We then move to the problem of Word Sense Disambiguation where the sense of the word is to be disambiguated based on the context in which it appears. Parsing is another problem that has utility in most other applications. Here, we are needed to decipher the parse tree corresponding to each sentence. Finally, we shall look at Machine Translation where we are required to translate a sentence in one language to another.

2 Spell Check

In the problem of Spell Check, we are required to detect mistakes in the given text and provide suggestions. Mistakes can be of two types - context free and context

sensitive. Context free spell check focusses on correcting mistakes without looking at their context. So, the only mistakes that can be detected by this method are those words that are not present in the dictionary (i.e) words with spelling mistakes. Context sensitive spell check, on the other hand encompasses methods that perform spell check by looking at the context words. So, this method can be used not only to correct words that are not present in the dictionary, but also the words that are used in wrong context. For example, consider the sentence "Camels live in desserts". "Desserts" in this sentence is a context sensitive error which should be replaced by "desert"

2.1 Context Free Spell Check

Context Free Spell Check is implemented by the idea of "Noisy channel model" which is motivated from Communication theory. The basic framework of Noisy Channel model is that a signal from transmitter propagates through a communication channel and is observed at the receiver. The received signal gets deteriorated due to the effect of Noise in the channel. There are many ways of modelling the added noise, one popular approach is by considering it as Additive White Gaussian Noise. The signal is recovered by making use of the Noise distribution information as follows:

$$P(r|e) = \frac{P(e|r)P(r)}{P(e)} \tag{1}$$

Here, r represents the received signal and e represents the transmitted signal. This equation follows directly from Baye's Rule

Medical diagnosis is another field where Noisy channel model is very effective. Denoting c as cause and s as symptom, the equation for diagnosis is written as

$$P(c|s) = \frac{P(s|c)P(c)}{P(s)} \tag{2}$$

A similar idea could be used for Context Free Spell Check. Let us say, we detect a word as a typo. This can be detected by searching the dictionary and checking if the word is found. Efficient search could be performed by hashing. After a word is detected as typo, we make relevant suggestions as follows:

$$P(c|t) = \frac{P(t|c)P(c)}{P(t)} \tag{3}$$

Here, c denotes the candidate suggestion word and t denoted the typo. By computing the posterior for all candidate words, we can make the best suggestion by taking the maximum posterior value. It is important to note that the term $P(t)$ in the above equation need not be estimated since it is common for all the candidate words. There are many ways to compute the likelihood term $P(t|c)$, one way is by using edit distance between the words t and c. $P(c)$ indicates the prior term which denotes the probability that the word c is present in

the dictionary.

To account for new words which are not yet encountered, we add a smoothing penalty (ie)

$$P(c) = \frac{n + 0.5}{N + 0.5v} \tag{4}$$

2.2 Context Sensitive Spell Check

In this task, we are required to correct mistakes based on the context, (i.e) a word may be a valid one, but its improper use in context would will make it erroneous. To detect and correct errors of this kind, it becomes essential that we look at the context around the word rather than the word itself. Two popular features used for this task are the Context words and Collocations.

Context words look at the words around a target word to decide if the word is erroneous. For instance, if the neighbourhood around a word contains words like "arid", "hot", etc. it is most commonly "desert" than "desert". To perform Context sensitive spell check, we store the set of confusion lists, which are the most commonly confused words. Some examples include desert, dessert, cite, sight. A word found in any of the confusion set is a target word that needs to be checked. We look at +- k words around a target word and store these as features. Now, we compute the posterior for each word in the confusion list as follows:

$$P(w|c_{-k}....c_k) = \frac{P(c_{-k}...c_k|w)P(w)}{P(c_{-k}....c_k)} \tag{5}$$

By similar argument made before, we can ignore the denominator in the above expression. Moreover, we can also make the conditional independence assumption (i.e) context words are independent given the target word. This assumption is similar to Naive Baye's idea in Machine Learning. The posterior finally reduces to

$$P(w|c_{-k}....c_k) = P(c_{-k}|w)...P(c_k|w) \tag{6}$$

Each of the likelihood terms can be learnt from the data as the ratio of number of words for which the context word c_i is within +-k words around the target word to the total number of occurrences of the target word. We can also incorporate smoothing term as discussed in the previous section.

Since the context words treats the context as bag of words, it loses the sequence information. To take into account the sequence, we use collocation features. In simple terms, collocations can be mentioned as the sequence of parts of speech tags of the neighbouring words. Typically, the words in the confusion set will co-occur with different sequences of part of speech words and so this becomes a good feature to disambiguate between words in a confusion set. A

very similar posterior computation as context words can be made for collocation, except that we can't make Naive Bayes assumption here.

Context words and collocations are two sets of features that are successful in spell check applications. Instead of using them separately, we can together use both features to get best results. Decision lists are the popular classifiers that have shown great success for this task. We can use other classifiers as well.

2.3 Evaluation Metrics

Many evaluation metrics can be used for Spell Check. One way is to use word error rate, which can be mentioned as the ratio of wrong corrections by total number of mistakes. But this will result in severe skew in estimate as the number of errors are typically less. Other way to circumvent this issue is by using a trigram metric, where the letterwise trigrams between the suggested word and reference word can be computed and the ratio of correct trigrams can be reported.

3 Information Retrieval

The next task in our discussion is the problem of Information Retrieval. This problem is encountered in our everyday lives in search engines where we demand a set of documents to be retrieved based on our query. Essentially, we are required to compute relevance the between a query and a document.

One way of formulating this task is using the notion of our familiar Noisy Channel Model. Here, the query can be treated as the cause and the results as the symptom. But this doesn't fit well into the picture because the cause is our intent which can't be represented well. Instead we use utility as a proxy.

One of the most successful methods for Information retrieval is the vector space model. The idea of this model is to represent the documents as a vector of words. Now, measuring the similarity between any two documents boils down to measuring similarity between these vectors, which can be done by metrics similar to the cosine of the angle between these two vectors. A query can also be treated as a document and using this similar documents can be extracted.

Now let us look at how to represent the documents as vector of words. To represent any document, we need to have the number of dimensions of the vector equal to the vocabulary size. One naive way of representing the vector would be to assign as components of these vector the term counts of the corresponding word in the document. But, this is a local measure and doesn't take other documents into account. One very popular way of forming these word vectors is by using $tf - idf$ scores. Here, tf stands for term frequencies and idf stands for

inverse document frequency.

$$tf - idf = tf.\log\frac{n}{N} \tag{7}$$

where tf stands for term frequency, n is the total number of occurrences of the word in all documents and N is the total number of words. The tf-idf weights encompass both local and global weights, thereby giving better estimates.

3.1 Evaluation Metrics

Precision and recall are the most common metrics used in most of the machine learning tasks. This can be used for Information Retrieval systems as well. Precision and Recall metrics are defined as follows:

$$\text{Precision} = \frac{Relevant \cap Retrieved}{Retrieved} \tag{8}$$

$$\text{Recall} = \frac{Relevant \cap Retrieved}{Relevant} \tag{9}$$

But the problems with these measures are that they don't account for ranking and degree of relevance. So, a better metric would be the Mean Average Precision. Let us denote the array of suggestions as $s[1..n]$. Then, mean average precision is defined as

$$\text{Mean Average Precision} = \frac{Precision(s[1]) + Precision(s[1..2]) + ..Precision(s[1...n])}{n} \tag{10}$$

There are some potential shortcomings with information retrieval systems. First, they consider a bad of words representation and so, sequence information is lost. Second, since the intent is not captured well, we use utility as a proxy to model intent. Moreover, we are also a victim of curse of dimensionality due to very large size of the vocabulary.

4 Estimating Word Similarities

One sub problem that arises from the Information Retrieval systems is estimating the word pair similarity. We utilize knowledge from several sources for this. Three of the most common categories of knowledge acquisition for this task are Linguistic knowledge, External world knowledge and Introspective knowledge. We shall look at each of these methods below:

4.1 Linguistic Knowledge

We use the linguistic knowledge stored in WordNet, which is the lexical database for English knowledge. Here, each word is associated with a synset(which represents a distinct concept). In order to estimate the similarity between two words, we can estimate the similarity with respect to their respective synsets. WordNet gives a hierarchical structure for various words, and to estimate the similarity between two words, we can look at path based closeness in the tree, or we can use information theoretic measures based on entropy.

4.2 External Sources

Information from external sources can also be used to estimate the word similarities. Wikipedia is one of the largest sources of knowledge available till date. So, Wikipedia concepts can be used to estimate the similarity between words using a technique called "Explicit Semantic Analysis"

In this method, we first get the $tf - idf$ vector corresponding to a Wikipedia concept. We can also get an inverted index, where each word can be represented as a vector of concepts. Now, to estimate the similarity between two words, we take the cosine of the angle between their corresponding concept vectors. We can also implement an information retreival system using this method. The given query can be treated as a document and its word vector can be generated. Now, by choosing the concept closest to this vector, we can get the relevant documents.

4.3 Introspective measures

Introspective/ Distributional methods are based on the premise that words are usually characterized based on their neighbours. So, these methods use the idea that two words are similar of they have the same context words around them. So, we can look at a k-window around each of the word and represent the word as a vector in context word space. Instead of Word-Document matrix that we constructed so far, we construct a Word-Context word matrix now.

Again, to estimate the similarity between two words, we could potentially used a cosine metric like the one used in the methods discussed above. But for this task, it has been shown that pointwise mutual information between the distribution of context word around two target words are a better indication of the similarity. We could also use KL- divergence between the two distributions to infer their similarity.

All the three methods discussed above use explicit knowledge by some means. WordNet based method use the knowledge of the hierarchical tree bank that encodes relations between object categories to infer the similarity between words.

Wikipedia based measures, on the other hand use the knowledge of the category wise sorted articles to estimate the similarity. Distributional measures are similar to Wikipedia based measures but instead of looking at words clustered as concepts, they look at words clustered around a small neighbourhood. So, Wikipedia based measures and distributional measures represent different granularity of implementation of the same idea.

5 Word Sense Disambiguation

Word Sense disambiguation is the task of disambiguating the sense of a word given a context. Many of the words are polysemous, the sense of the word could only be deciphered in the presence of context in which they occur. For instance, consider "I ate an apple" and "Apple is expanding its markets in India". In the first sentence, we talk about the fruit sense of the word Apple, but the second sentence talks about the company "Apple".

This task can be seen in some sense as converse to estimating word similarities. Here, instead of estimating similarities, we are required to broaden the gap between two senses of a word. A simple model for WSD could be the famous Noisy channel model where intended meaning could be seen as the cause and the word form as the sense. We can also use WordNet to estimate scores for each sense based on path based similarity with the context words. The sense with the maximum score can then be chosen as the desired sense of the word.

In most of the works, Word Sense Disambiguation is posed as classification problem. It can be posed as supervised, unsupervised or semi-supervised learning task, with supervised being the most common. Features are extracted and classifiers are trained on a suitable corpus. Nearest Neighbours, Decision List, Neural Networks and Support Vector Machines are some of the most popular classifiers used for this task. However, most of the time, we don't end up getting a good corpus. So, recent approaches have focussed on using semi-supervised or unsupervised approaches for this task.

5.1 Evaluation

Word Sense Disambiguation is not very useful as a standalone module, but it is a basic step in most of the other NLP tasks. So, instead of evaluating WSD alone, it makes more sense to evaluate this with the aid of some other task. This can be done by comparing the performance of a task with and without the presence of WSD system.

6 Circularity

One of the common issues with most of the NLP tasks is the circularity. In Information Retrieval Systems, we are required to estimate the Word-Document

similarity matrix. The circularity encountered here is as follows: Two words are similar if the documents in which they occur are similar and two documents are similar if they contain the similar words. It is essential that we break this circularity. In the next few sections, we discuss various topics where we encounter such circularity and look at how we break them.

6.1 Page Rank Algorithm

Page ranks are the ranking that are made for each of the web page and are used for efficient document retrieval in search engines. The basic idea behind the Page Rank Algorithm is that "A page is important if it is linked to by several important pages". This definition directly leads us to the circularity as follows: A page is important if it is linked by several other pages, which in turn are important if they are linked by other pages which can include this page.

The circularity is resolved as follows. Let us say that the score corresponding to page i is p_i. Let α_{ij} represent the binary variable denoting the links, (i.e) $\alpha ij = 1$ indicates that page i is linked to j. Using this we can write the equations as follows:

$$p_i = \sum_{j=1}^{N} \alpha_{ji} p_j \tag{11}$$

Denoting P as the vector $< p_1 ....p_N >$, the set of equations can be written in matrix form as $P = \hat{\alpha} P$. This can be solved using Eigen vectors and so the circularity is resolved.

6.2 Latent Semantic Analysis

This is another method for breaking the circularity of Word-Document Similarity. Instead of mapping words to document, this method introduces a temporary layer in-between where latent concepts are learnt. In Explicit Semantic Analysis, words were mapped with concepts using Wikipedia, which is human annotated. But, in this method, we try to introduce the concepts implicitly.

The method is based on the idea of Singular Value Decomposition. The given Word-Document matrix is decomposed into three matrices(according to SVD), where the first one is column orthogonal, second is diagonal and the third one is row orthogonal. Now, the first matrix can be thought of as Word-Concept matrix, and the third one is Concept -Document matrix and the second one just performs scaling. We can also reduce the dimensions by choosing one top k singular values and this could improve the performance in some applications.

6.3 k-means Clustering

Clustering is a standard machine learning problem, where a group of objects are required to be clustered without supervision. This is a hidden variable problem,

since the information about the classes of objects are hidden. This problem is grounded on the circularity that " A point belongs to a cluster if it is closest to the mean of a cluster, and the mean of a cluster depends upon points belonging to the cluster".

We can resolve this ambiguity using "Expectation Maximization" algorithm. EM trains the model by iterated updates, where a model is fixed first which is used to fit the points, and the fitted points are then used to update the model parameters. In k-means clustering, the cluster centres are fixed first using which clusters are assigned, and the assigned clusters are then used to update the cluster centres. This process is repeated over multiple iteration until convergence.

7 Statistical Parsing

The problem of parsing can be stated as the task of obtaining the parse tree corresponding to an input sentence. The lowest granularity in parsing is reaching atomic words and assigning part of speech tags to each of them. So, part of speech tagging is a sub-problem of parsing.

Once we have a Part of Speech tagger system, we need to then get parse trees corresponding to a sentence. To do this, we need to define grammar corresponding to our language. Grammars can be categorized as Regular Grammars, Context Free Grammars, Context Sensitive Grammars and Recursively Enumerable Grammars. Most common grammars used for English language is the Context Free grammar.

We model the parsing problem in a statistical framework, where each of the grammar rules are assigned a probability. So, for a sentence, there can be multiple parse trees and each of them will have a probability associated with them. It becomes necessary to estimate these probability values in order to make an inference given a target sentence.

7.1 Expectation Maximization for Probabilistic Context Free Grammars

Estimating the parameters of PCFG is hard because we don't have the probability of grammar rules, and we also don't have the assignment of the sentence to the parse tree. We can train the model using Expectation Maximization Algorithm. As an initialization step, all grammar rules are assigned equal weights. Using this, probabilities associated with each of the parse tree is computed and the assignment of sentence to the parse tree is made. Further, this with the help of these assignments, the rule probabilities could be learnt. This process iterates till convergence.

8 Machine Translation

Machine Translation is a problem where given a sentence in one language, we are required to translate it to another language. This problem is hard because the grammar rules in one language is different from another and so, a simple word by word translations will most likely fail. So, for each sentence , the alignment information that exists from one language to another needs to be captured. In the parallel corpora, we won't have this alignment information. So, alignments can be seen as hidden variables.

Expectation Maximization can once again solve this problem. In one step, given the word alignments,we can learn the translation probabilities. Then, by fixing translation probabilities we can learn the alignments. This process can be repeated till convergence. While testing, we need to adopt a generative approach, where a new set of alignments are generated for each sentence, and based on the translation probabilities we can get the best possible sentence.

However, there are many problems with this approach. First, the length of sentences in both languages need not be equal. Second, one word can be translated to multiple words in other language, or new words can be generated in the second language. So, we need to take care of all these issues while training.

9 Natural Language Generation

Till now, we have discussed about various problems in Natural Language Understanding. Now, let us look at the other dimension of NLP - Natural Language Generation. Natural Language Generation systems are needed to generate information, just like humans generate text. One example of such system is reading weather statistics and generating a summary in the form of a report. NLG systems can also be used in Medicine as Medical Prescription Generation machines.

10 Conclusion

In this chapter, we discussed about the key concepts in NLP. We also looked at several problems and their solution techniques. Ideas inspired from diverse fields were highlighted and their applications in NLP were mentioned. Fundamental limitations of each of the techniques were also discussed and connections between different topics were made.

References

Class lecture notes

.Jurafsky and J.H. Martin *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*

Chapter 4

Yogesh Balaji

Indian Institute of Technology, Madras
ee12b066@ee.iitm.ac.in

Abstract. This chapter presents a summary of the paper, "Minimum Error Rate Training in Statistical Machine Translation" by Franz Josef Och. One of the main problems in many of the Statistical Machine Translation methods is that there is a loose relation to the final translation quality on the unseen test data. This is because the training criteria and final translation evaluation criteria are different. This paper presents a novel algorithm for efficient training using the criteria that directly optimize the translation quality. They show that significantly better results are obtained if the final evaluation criterion is taken into account as a part of training procedure.

Keywords: Minimum Error Rate Training, Statistical Machine Translation

1 Introduction

In Statistical Machine Translation, the evaluation metrics used are beyond the simple Word Error Rate criterion, which simply counts the number of errors with respect to a reference sentence. Some of the most popular metrics are the BLUE score which compute the geometric mean of n-gram precisions of various lengths and the NIST score which computes a weighted average of n-gram precisions. Most of the machine translation models, on the other hand use maximum likelihood or related criterion for training. So, there is a mismatch between the basic assumptions of the statistical models being trained and the final evaluation criterion. Ideally, we would require our system to train using the same testing criterion for better performance. This paper presents an algorithm for efficiently optimizing model parameters with respect to any training criterion.

2 Log-Linear Models for Machine Translation

Let us assume that we require our French sentence $f = f_1, f_2...f_J$ to be translated to English sentence $e = e_1, e_2...e_l$. Using a log-linear model, the posterior probability is given by

$$P(e|f) = p_{\lambda_1^M}(e|f) = \frac{\exp \sum_{m=1}^{M} \lambda_m h_m(e, f)}{\sum_{e'} \exp \sum_{m=1}^{M} \lambda_m h_m(e', f)} \quad (1)$$

Here, λ_m are the model parameters and $h_m(e, f)$ are the set of M feature functions. In the training process, we need to obtain the parameter values λ_1^M. A standard criteria for Log-Linear models is the Maximum Mutual Information criterion, which gives:

$$\hat{\lambda_1^M} = \underset{\lambda_1^M}{\operatorname{argmax}} \sum_{s=1}^{S} \log(p_{lambda_1^M}(e_s|f_s)) \tag{2}$$

This optimization problem is preferred because it has one unique global optimum and there are efficient algorithms for convergence. Unfortunately, this doesn't take into account the test error criterion and so, it doesn't guarantee good performance on test set. In the next section, we shall see how to train the given model for any given error criterion.

3 Training Criteria for Minimum Error Rate Training

Let us assume we have an error function E(r,e) that quantifies the error between the candidate sentence e and the reference sentence r. We shall also assume the error for S sentences is additive. To put it mathematically,

$$E(r_1^S, e_1^S) = \sum_{s=1}^{S} E(r_s, e_s) \tag{3}$$

Let us say that our translation system provides us with a set of K different candidate translations $C_s = e_{s,1}...e_{s,k}$ for each input sentence f_s. We will pick the best candidate translation according to (1) and use this to optimize our parameters.

$$\hat{e}(f_s; \lambda_1^M) = \underset{e \in C_s}{\operatorname{argmax}} \sum_{m=1}^{M} \lambda_m h_m(e, f) \tag{4}$$

$$\hat{\lambda_1^M} = \underset{\lambda_1^M}{\operatorname{argmin}} \sum_{s=1}^{S} E(r_s, \hat{e}(f_s; \lambda_1^M)) \tag{5}$$

This optimization is hard because it contains an argmax, due to which gradient can't be computed. Moreover, the objective function also has many different local optima. One way of resolving gradient computing problem is by computing a smoothed error function that weighs all candidate suggestions by their likelihood scores.

$$\hat{\lambda_1^M} = \underset{\lambda_1^M}{\operatorname{argmin}} \sum_{s=1}^{S} E(r_s, e_{s,k}) \frac{p(e_{s,k}|f)^\alpha}{\sum_k p(e_{s,k}|f)^\alpha} \tag{6}$$

This is called the smoothed estimate of error criterion. In this paper, they have shown that both smoothed and unsmoothed method of training gives similar performance. In the next section, we shall discuss efficient optimization algorithm for unsmoothed error function.

4 Optimization Algorithm for Unsmoothed Error Count

One way of optimizing (5) is by using line search, where we start with a random value assigned to the parameters and move along one direction each time, (ie) make one dimensional update while keeping all other dimensions constant. This algorithm is very slow and we seek faster and stable optimization methods than line search.

The main idea proposed is to approximate the error surface by a piecewise linear function that enables efficient traversal to determine the optimal parameters. Assume, we make parameter updates along the line given by $\lambda_1^M + \gamma d_1^M$. Then, computing the most probable sentence (equation (4)) reduces to

$$\hat{e}(f_s; \lambda_1^M) = \underset{e \in C_s}{\operatorname{argmax}} \sum_{m=1}^{M} \lambda_m h_m(e, f) + \gamma \sum_{m=1}^{M} d_m h_m(e, f) = \underset{e \in C}{\operatorname{argmin}} (t(e|f) + \gamma m(e|f)) \tag{7}$$

where t and f are the negative of the respective terms. Let us represent f as

$$f(\gamma; f) = \min_{e \in C} (t(e|f) + \gamma m(e|f)) \tag{8}$$

The function $f(\gamma; f)$ is piecewise linear as t and m are constant with respect to γ. Let us represent the piecewise linear boundaries by $\gamma_1^f < ... < \gamma_{N_f}^f$. Let us also store the changes in error count in each segment $\Delta E_1^f, ...\Delta E_{N_f}^f$. Here, ΔE_n^f is the difference between the error count at $\frac{\gamma_n^f + \gamma_{n+1}^f}{2}$ and $\frac{\gamma_{n-1}^f + \gamma_n^f}{2}$. We can compute γ^f and ΔE^f for every sentence in the corpus. The optimal value γ can me computed by traversing the boundaries while updating the error count as the optimum optimum points always lies in the boundaries (from the theory of Linear Programming)

We can extend the proposed algorithm for BLUE and NIST score by changing the error function appropriately.

5 Experiments and Discussions

The basic translation framework used is segmenting the test into phrases, translating individual phrases and then reordering them to generate the final sentence. The feature functions $h_m(e, f)$ used include phrase penalty and spatial alignment features. To generate a set of K candidate translations, a dynamic programming beam search algorithm is used to explore a subset of possible translations, followed by pruning using A_* search. Since training on these K candidate translations alone won't perform well, the first K set of sentences are used to train the model. The trained model is then used to perform a new search to get the new K set of sentences which is then merged to the old set. This process is repeated

until no new set of sentences are generated.

Another issue in applying Maximum Mutual Information criteria as discussed in Section 2 is that the K best list generated need not include the reference sentence. So, the pseudo reference sentence is made, wherein the sentence best in the set of K sentences based on (4) is chosen to be reference sentence. Due to this, there is bias of this method towards Maximum Error rate criterion.

Experimental evidence indicates that their method of training gave good performance on different error criteria used. Moreover, they also indicate that when the system is trained using one error function, it performs the best when tested on the same error criterion compared to other criteria.

There were some details that I didn't understand well in this paper. First, I am not very clear about how the set of candidate translations are generated. It is mentioned that they used beam search and A^* pruning, but it wasn't mentioned what was it performed on. Second thing that is not clear is the system for machine translation. The link between the feature functions and translation system is not established well.

6 Conclusion

This chapter explains how the training could be performed using different error metrics that are used for testing Machine Translation systems. Incorporating these metrics inherently improves the translation quality as the parameters of our model will be tuned appropriately.

References

Och, Franz Josef, *Minimum Error Rate Training in Statistical Machine Translation*, ACL, 2003.