Observation and Modeling of Structures in Natural Language

Dietrich Klakow
Philips Forschungslabor
Weisshauststr. 2
D-52066 Aachen
Dietrich.Klakow@philips.com

ABSTRACT

Observing natural language to understand its underlying structure is essential for many applications like speech recognition, machine translation, information retrieval and spoken dialogue systems. Very often the structures that intuitively seem to govern natural language are not the ones that carry a lot of information (as measured in bits in the sense of the Shannon game). Huge corpora are now available to verify ideas about language and to reliably learn the laws of language (in the sense of the previous sentence) from data without bias. During the last years the topic of capturing long range dependencies in language was central in the community and various approaches are being developed that will be briefly described.

1. INTRODUCTION

Understanding of natural language is important from a scientific point of view. There are two basic approaches. The first is linguistically oriented and focuses on the special phenomena of language. It provides structural knowledge about language sometimes based selected examples. The second one is statistical and uses huge collections of text to make observations. Both approaches have produced a plethora of results.

This paper will mainly focus on the statistical corpus based approach. Huge text collections (so-called corpora) are now available for many languages. The largest corpus is probably the North American News Text Corpus with over a billion words from various newspapers. Such text collections can be used to determine statistics, to derive models from them and thus learn the basic laws of language. Corpora can also be used to verify ideas which were developed by some other means. The advantage of corpora is that they contain linguistic phenomena according to their relevance. That ensures that the applications built by a corpus-based approach will often work very
well as long as the data used to derive the models is similar in topic and style to the texts to expect in the application.

Some of these applications will be introduced in this paper to motivate the work on natural language. The paper will continue with an introduction of the basic notions of language modeling and a simple statistical law (so-called Zipf's law) that is valid for basically any corpus. We then continue to one of the essential modern topics: long-range dependencies in language from a linguistic and a statistical point of view.

2. SOME APPLICATIONS OF NATURAL LANGUAGE PROCESSING

Among the many applications, two areas will be focused on in this paper: automatic speech recognition and knowledge management.

2.1. Speech Recognition

Speech recognition [1] is the task of converting a speech signal into a word sequence. The basic architecture of a speech recognizer has been well established for a couple of decades now. Fig. 1 shows the basic components.

The feature extraction has to convert the time continuous signal into a sequence of feature vectors. Its task is to keep the essential information about human language but at the same time removing noise and speaker variability (e.g. average pitch). Those two goals are contradictory, which makes the task very challenging. A typical feature extraction produces a feature vector every 10 milliseconds.

A speech recognizer uses two probabilities. The first one is called the acoustic model and relates the

![Figure 1: Basic structure and components of a speech recognizer](image-url)
feature vectors $A$ of a set of training utterances to the corresponding words $w$. As the training set is usually not large enough to cover all possible words, a lexicon as an intermediate data structure is used. There the relation between words and phonemes is defined and only the phonemes have to be observed in the training set.

The second model is called the language model. This is the data structure which covers syntactic and semantic constraints of the language. A better understanding of language helps to improve this component of a speech recognizer. So far, it has turned out that the best approach to learning the structure of language is based on observing corpora and deriving model structures from them. These models have free parameters which are usually trained on collections of texts (like newspapers). The search has to identify the optimal word sequence corresponding to the sequence of feature vectors. It can be shown that the number of incorrectly recognized utterances is minimal when the word sequence is determined that maximizes the product of the probabilities corresponding to the language model and the acoustic model. The search procedure of a speech recognizer usually starts with the beginning of a sentence. It then tries to guess the next word given the next uninterpreted feature vectors and using the previously recognized words.

Here is an example:

```
" What’s in your hometown newspaper  ???"
```

The task of the speech recognizer is to complete the sentences and to determine the word that best replaces the “???” taking both the speech signal and the previous words into account.

## 2.2. Knowledge Management

Knowledge management is a seemingly unrelated field. It is also a field which has been evolving rapidly during the last years and in which many new companies have been founded. However, the field lacks a very clear definition and this paper can only give examples of sub-fields.

### 2.2.1. Topic Detection

Imagine a newspaper without any structure, just a collection of articles in random order. The task of topic detection is to identify the genre of an article, like “Business News”, “Politics” etc.. To do this automatically, the computer has to “understand” the articles. However, it turns out that a deep understanding is not necessarily needed. Very good results can already be obtained by only looking at the frequency of content words and comparing these frequency distributions with the frequency distributions of texts about a known topic.

### 2.2.2. Information Retrieval

Huge text collections exist and probably the largest one is the internet. The task of information retrieval [2] is to find the text that best matches a user query, which is given by a set of key words. Search engines in the internet are a typical application of information retrieval. At first sight this task has nothing to do with the previous one. However the algorithms are closely related. Again the frequency of the content words on the user query and the texts in the document collection are calculated and are matched against each other.
2.2.3. Question Answering

Question answering is a refinement of information retrieval. Specifying a query by a couple of content words is a very tedious approach. Ideally we would just like to ask a question like “What is the name of the capital of Nepal” and the search engine should present the section of a text from the collection that answers it. It is easy to see the relation to speech recognition. If we rephrase the question as “The name of the capital of Nepal is ???” and compare it to the example in the section on speech recognition we see that question answering may be nothing but speech recognition without a speech signal. Hence the two tasks may be related, and probabilistic approaches may also be applicable to question answering.

3. ELEMENTS OF LANGUAGE MODELING AND SIMPLE OBSERVATIONS ON CORPORA

3.1. Elementary Definitions

In the previous sections we looked at the application of language models, such as helping a speech recognizer to distinguish the acoustically similar sentences “eat the sandwiches” and “eat the sand which is” based on the probability of a sentence. So far we used P(W) to denote the probability of a sequence of words $w_1, \ldots, w_n$. However, as already indicated in the description of the search procedure of a speech recognizer, a time-synchronous approach is much better suited. To this end the probability of a sentence has to be decomposed. The definition of conditional probabilities can be used to factor the probability of a sentence into the probability that a word follows a given beginning of a sentence. Mathematically that is $P(W) = P(w_1, \ldots, w_n) = P(w_1) P(w_2|w_1) \ldots P(w_n|w_1, \ldots, w_{n-1})$. Hence the task of language modelling is to determine the probability $P(w_i|w_1, \ldots, w_{i-1})$ based on a collection of text.

There is a well-established measure for the quality of a language model. It is called perplexity [3] and is defined by

$$PP := P(w_1, \ldots, w_n)^{-\frac{1}{n}}$$

It has a very simple interpretation. In the section on speech recognition we looked at the task of completing a partially recognized sentence. Perplexity tells us how many times we have to guess a word by just looking at the predecessor words and ignoring the speech signal until we guess the right one. Note that perplexity is a measure “on average”. The game of completing a sentence has to be played many times on a large number of sentences. Only then does the number of trials match perplexity.

3.2. Zipf's Law

As an introduction to the corpus based approach, which tries to learn from corpora, Zipf's law will be discussed. It is an example of a very simple relationship that can be derived from many texts and is rather surprising. The recipe is simple:
• A word is everything between blanks
• Count the frequency for all words
• Sort this list by frequency
• Plot the frequency versus the position in the sorted list (the rank) on a double logarithmic scale

A typical result is shown in Fig. 2. A clear power law behavior is observed. This is mathematically best described by the Mandelbrot function [4]

\[ N(r) = \frac{1}{(c + r)^{\alpha}} \]

where \( r \) is the rank and \( N(r) \) the frequency. The function has two free parameters: \( a \) and \( c \). Only these vary for the different corpora. To show the flexibility of Zipf’s law the figure gives results for the Russian novel “Crime and Punishment” and a C-program (the Philips research speech recognizer). Zipf’s law is not only valid for all natural languages but very often also for other symbol sequences like music and DNA.

The knowledge of Zipf’s law can be used in applications where the frequency of words is essential. In those cases it can be used to reduce the number of free parameters of a model.

4. OBSERVING AND MODELING LONG-RANGE DEPENDENCIES

A typical problem with present speech recognizers is that they produce output that does often not make sense when looking at the complete sentence. Here are two examples:

• “A corporate and municipal bond prices also slumped.”
• “The October rise was initially reported as zero point two percent drop”
In the first example the sentence is correct until the word “prices”. It contradicts the word “A” (at the beginning of the sentence), which was incorrectly recognized (a so-called insertion). The second sentence is correct until the word “drop” occurs. Now it becomes obvious that something is missing. “Drop” requires an article, which in this example would be an “a”. The speech recognizer “deleted” the word “a” between “as” and “zero”. Those two examples show the typical problem. So far two different approaches have been taken in the literature to ensure the consistency of a sentence. Both will be briefly described in this article: grammars and log-linear models.

4.1 Grammars

Grammars are a well-established tool in the computer linguistic community to obtain a syntactic structure for a sentence. However, probabilistic grammars can also be used to determine the likelihood of a sentence. In Fig. 3 we see the grammatical analysis of the correct sentence (left) and the sentence that was recognized using a standard speech recognizer. It is easy to see that the grammatical analysis of the incorrect sentence detects that the verb is missing. This makes the sentence very unlikely. We use the parser of E. Charniak [5] and combined its output with a traditional language model. This resulted in an 8% improvement in recognition result showing the benefits of combining a powerful grammar with a backing-off trigram [3].

4.2 Pair-Correlation Function and Log-Linear Interpolation

A very simple way to measure long range dependencies on a corpus is the pair correlation function. For simplicity we only consider the auto-correlation function which is defined by:
where $P_d(w \ w)$ measures the probability that a word re-occurs after $d$-words in between. Any sequence of words may fill the gap between the two occurrences of $w$. The denominator has a very simple motivation. As $d$ tends to infinity the probability $P_d(w \ w)$ tends to $P(w)P(\ w)$. Hence the pair correlation function tends to 1 for very large distances. Fig. 4 shows the pair-correlation function for four English words from the Wallstreet Journal Corpus. We see that each word has its own typical pattern. At short distances we observe very sharp peaks reflecting grammatical constraints or frequent phrases. In the intermediate range semantic coherence dominates the curves (e.g. “president”). At large distances ($d>1000$) we observe statistical independence coming from the fact that successive newspaper articles in the Wallstreet Journal are basically uncorrelated.

The pair-correlation function can be used to improve a baseline language model further. A strict mathematical derivation is possible but we focus here on an intuitive approach. A pair-correlation function larger than 1 indicates that this word is likely to reoccur. A pair correlation function smaller than 1 shows that the likelihood that the word is the next one in the sentence should be decreased. This behaviour can be modelled by the so-called log-linear interpolation [5]:

\[
P(w_n \mid w_{n-1}w_{n-2}…w_{n-1+d}) = \frac{1}{Z(h_{df})} P(w_n \mid w_{n-1}w_{n-2}) \prod_{d=1}^{M-1} \left( \frac{P_d(w_n \mid w_{n-d})}{P(w_n)} \right)^{\lambda_d}
\]

In the brackets the pair-correlation function is given in a different notation using the definition of conditional probabilities. The exponents $\lambda_d$ weight the different contributions. Typically the
exponents take values around 0.5. The product finally ensures that all possible distances are taken into account. This product modifies the baseline model. A model of this kind is not necessarily normalised (i.e. the probability doesn’t sum to one). To this end the explicit normalisation term \( Z(h_M) \) is introduced. Log-linear interpolation of language models as been used successfully to build models which cover 20 words, thus ensuring semantic coherence over this range. Perplexity reductions of up to 30% have been measured.

5. CONCLUSION

This paper tried to introduce the reader to a corpus-based approach to modeling natural language. The applications of natural language processing are manifold. We illustrated some of them. Language models assign probabilities to sentences. The goal is to model complete sentences. Two approaches were mentioned:

- Grammars
- Pair-correlation functions and log-linear interpolation

First experiments on combining linguistic and statistical knowledge are very promising. Future investigations should focus on a deep understanding of the combination of both paradigms.

6. REFERENCES