Natural Language Processing - Study Notes

Eduart Uzeir

This paper is the product of my study for the course of Natural Language Processing taught by professor Fabio Tamburin at Univeristà degli Studi di Bologna in the academic year 2018-2019. The material is organized based on professor's slides and lectures. Furthermore, additional learning material is taken by the books and online resources the prof. suggested during the lectures. It is also easy to find on-line courses and YouTube videos that can be helpful to a better understanding of Natural Language Processing.

1 Introduction

In this chapter we are going to do a brief but holistic introduction on the Natural Language Processing. The early developments, Chomsky's revolution and contemporary corpus-based linguistics. Furthermore we are going to see some of the main methods and techniques used in language processing such as statistical language models and machine learning oriented techniques applied on language.

1.1 NLP - Overview

Computational Linguistics or Natural Language Processing is a highly interdisciplinary field that deals with language processing using computers. It gathers contributions from other fields like **linguistics, psychology, information theory, mathematics, statistics** and of course **computer sciences**. The conceptual model of NLP is shown in the following figure.

There are three main phases that we can study the evolution of the field:

Universita' degli studi di Bologna, e-mail: eduart.uzeir@studio.unibo.it

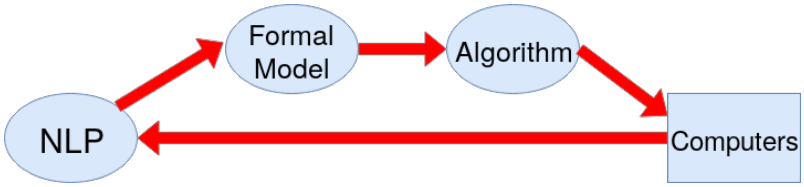

Fig. 1 Conceptual model of NLP

- **Empiricist** approach: is the study of linguistics based on spoken or written natural language. Is important to mention that the study of language those day was limited by the technology and the amount of data.
- **Rationalist** approach: Chomsky's work in 1957 revolutionised the field of linguistics. He proposed new ideas that changed the whole perception of how the language has to be studied. Here are the main ideas Chomsky introduced:

  1. Language is a **formal object** that can be fully described mathematically.
  2. **Introspection** is the main tool used to describe the language. This mean that a native can explore and describe the language directly without the need of using corpora.
  3. Chomsky's approach to the language is based on a **deductive** reasoning.
  4. **Competence** vs. **Performance**. Chomsky argued that **performance** or the act of using the language is a poor indication of the **competence**. In other terms a native speaker may have many ways to express something but not all of them are used in the performative action.
  5. Natural language is recursive, consequently infinite. We can never construct an infinite corpus, so the corpus based study of language is **incomplete** and **skewed**.
  6. **Frequency** is the only parameter taken in consideration in a corpus based empiricist study of language, meaning that a word or a linguistic phenomena is included only if it is used frequently. Chomsky proposed that instead of waiting for a word to appear i just ask myself using introspection.

- **Neo-Empiricist** approach: With the progress of the technology and the vast amount of data available the empiricist approach resurgence at the 90s. Nowadays almost all the field of Natural Language Processing is corpora driven. Here are some of the reasons why corpora are important regardless of Chomsky's critics:

  1. In some linguistic level i.e phonetic observations is the only choice we have. Imagine a baby trying to learn to speak, how can we study the competence of a baby? In this case only the performance matters.
  2. Corpus-based study of linguistics is quantitative. We can actually measure and analyse phenomena using mathematical methods, such as probability etc.

3. Frequency in large corpora is a good approximation of a phenomena. If the frequency is very low this mean that the phenomena is an exception rather than the rule.
4. In the empiricist approach the corpora used is of extreme importance. The size of the corpora should be very large in order to provide actual evidence of language phenomena. This observation is forced by the **Zipf's Law**.

Zipf's Law is an empiric observation that relates the rank of a word (in our case) with it's frequency.

$$F \propto 1/R, (\exists k, F * R = k)$$

The Zipf's law eventually leads to the **"Sparse Data Problem"**, meaning that even thought the corpora are very large more data is needed to predict and describe certain linguistic phenomena.

1.2 We need some Probability Theory

Now that we have quantitative data provided by the corpus, how we can use them in order to create a Language Model (LM)...but what is a LM?
A **Language Model** is a model that use probability theory in order to predict the probability of a word in a sentence given the probabilities of the previous words in that sentence. More formally:
"By interpreting text or any linguistic production as the result of a probabilistic/stochastic process the linguists can use a corpus as a sample of text to analyse the distribution of words and other linguistics phenomena in order to design a ***probabilistic model*** *of the language in the corpus."* Before diving into how this is achieved, let's refresh some probability concepts. **Random Event** is called an uncertain event that it's occurrence is unpredictable or predictable in some degree of uncertainty.
Random Process is a process that produces random events.
Example of a random events:

- outcome of a coin toss (H/T)
- the event that a name immediately follows the article "the" The House ...

We use probability to assign a numeric value between 0 and 1 to uncertain events. This is to say that the probability measure our degree of uncertainty of one event to happen. The probability of all events over Ω is called a *Probability Distribution*.

- Usually the probability is described as the outcome of an experiment i.e coin tossing.
- The set of all possible outcomes of an experiment is called **Sampling Space** Ω.
- Given an experiment with sampling space ω a **Random Event** is any *subset* of Ω.

- We have two types of events:

  1. Simple Events: the set contains only one simple result $\{T\}$ - coin, $\{4\}$ - dies.
  2. Complex Events: the set contain more than one event $\{4, 3\}$ - roll of two dies.

If all simple events over Ω have the same possibility to occur then we say that the probability of the event **A** is:

$$P(A) = |A|/|\Omega| \text{ :: Probability of an Event "A"}$$

What is the probability of getting two times 3 if we cast the die twice? To answer this question we have to consider the **Joint Event Probability**.

- e1 = "get 3 in the first cast" P(e1) = ?
- e2 = "get 3 in the second cast" P(e2) = ?

$$P(e1 = 3, e2 = 3) = P(e1) * P(e2) = 1/6 * 1/6 = 1/36$$

To recap: If **A** and **B** are two independent events then their **Joint Probability** is given by the formula:

$$P(A,B) = P(A) * P(B) \text{ :: Joint Probability}$$

A sequence of words in the text is not the result of series of random choices, instead the words in a text influence and condition the presence of other words. We can capture this aspect with the notion of **Conditional Probability**.
Conditional Probability: Given the fact that the event **B** has already occurred the probability of a new event **A** conditioned to **B** is:

$$P(A|B) = P(A,B)/P(B) \text{ :: Conditional Probability}$$

In the case **A** and **B** are independent $P(A|B) = P(A)$, meaning that **B** do not influence **A**.
Example: $P(dog|the) = P(the, dog)/P(the)$.
Bayes Rule:

$$P(A|B) * P(B) = P(A,B) = P(B|A) * P(A) ==> P(A|B) = \frac{P(B|A)*P(A)}{P(B)}$$

At this point that we revised some of the most important concepts of the Probability Theory, we want to know how we can apply this concept to our language models? The only quantitative data the we can get from the corpus is the *frequency* of a phenomenon. We have to find some way to *approximate* the frequency to the probability.
We define the **relative frequency** as:

$$f_r(A) = \frac{f(A)}{N} \text{ :: N is the number of all objects i.e words in a corpus}$$

Frequentist definition of probability: is the approximation of the probability of an event using the relative frequency on a very large number of experiments.

$P(A) = \lim_{N\to\infty} \frac{f(A)}{N}$ :: Approximation of P(A) using f(A)

The relative frequency of an event A give us an empirical estimate of the probability of A. This is done by observing the number of times A has occurred. The accuracy of this estimate depends dramatically on the number of experiments, and as we know **corpora** is our source of experiments.

Fig. 2 Approximation of probability using frequency

Regardless the many opportunities this approximation method offers there is one problem that we have to deal with: What happens if the frequency of a phenomena is **0**?! There are two solutions for this problem:

- Smoothing techniques: we use different methods of smoothing like, ”add-one smoothing”, ”Good-Turing smoothing”, ”Kneser-Ney smoothing” etc.
- Distribution approach: consider your events as a probability distribution i.e Normal Distribution. This distributions do not permit 0 probabilities.

For further study on Statistics and Probability Theory for linguists please refer to this paper: ”The Linguist’s Guide to Statistics - B. Krenn, Ch. Samuelsson”.

1.3 N-Grams

As we said previously the goal of a Language Model is to assign probability to a sentence. One method to do this is to use **N-grams**. What are N-grams?!
N-grams are the formalization of the idea of ”word prediction” with the use of probabilistic models. In other words the idea is to predict the word *N* using the previous *N-1* words. N-gram language model is one of the most important and vastly used language models.
Estimators like *N-grams* that assign a conditional probability to possible next words can be used to assign a joint probability to an entire sentence.
How does N-grams work?!

The goal is to compute **P(w—h)**, the probability of a word **w** given some history **h**. For example, given the history **h** = "its water is so transparent than", we want to compute the probability of the next word **w** = "the".

Let's consider a sentence $\bar{\omega}$ made of some words ω_i and try to compute the conditional probability, $P(\bar{\omega})$.

$$P(\bar{\omega}) = P(w_1 * P(w_2|w_1) * P(w_3|w_1,w_2),... * P(w_n|w_1,...w_{n-1})).$$
$$P(\bar{\omega}) = \prod_{i=1}^{n} P(w_i|w_1,...w_{i-1})$$

At this point we can use the conditional probability formula $P(A|B) = \frac{P(A,B)}{P(B)}$ and substitute the probabilities with the frequency approximations $P(A) \approx \frac{f(A)}{N}$ and obtain the following formula:

$$P(w_i|w_1,...w_{i-1}) = \frac{P(w_1,...,w_i)}{P(w_1,...,w_{i-1})} = \frac{f(w_1,...,w_i)/N}{f(w_1,...,w_{i-1})/N} = \frac{f(w_1,...,w_i)}{f(w_1,...,w_{i-1})}$$

Here $f(w_1,...,w_i)$ is the frequency of the token sequence $w_1,...,w_i$ in the considered corpus. In very large corpora is very difficult to compute the probability for an arbitrary number of words contained in a sentence, so we restrict this number usually to number like "ONE", "TWO", "THREE" and rarely bigger numbers. This are the N-grams where N is one of the numbers.

$$P(\bar{\omega}) = \prod_{i=1}^{n} P(w_i|w_1,...w_{i-1}) = \prod P(w_i) :: Unigram$$
$$P(\bar{\omega}) = \prod_{i=1}^{n} P(w_i|w_1,...w_{i-1}) = \prod P(w_i|w_{i-1}) :: Bigram$$
$$P(\bar{\omega}) = \prod_{i=1}^{n} P(w_i|w_1,...w_{i-1}) = \prod P(w_i|w_{i-2},w_{i-1}) :: Trigram$$

Using N-gram model we can compute the probability the most likely word following the sequence of words $\bar{\omega} = w_1,...,w_{n-1}$ using the following formula:

$$\bar{\omega} = argmax_{w_n} P(w_n|w_1,...,w_{n-1})$$

What is *argmax*?

In mathematics *argmax* or *arguments of the maxima* are used as an abbreviation to indicate to point on the domain of a function that maximize the function's value. *argmax* refers to the input or arguments of a function at which the function output is as large as possible.

In our case, when we want to predict the probability of a word knowing the previous words we are interested on those parameters that maximize the probability of the predicted word.

Another very important Language Model is constructed using **Recurrent Neural Networks** called **RNNLM**. This new approache was predsented by T. Mikolov[1] et al. in a paper entitled ”Recurrent neural network based language model”. They achieved very good performance using variable size context input in a simple Elman network. The structure of their network is shown in the figure 3.

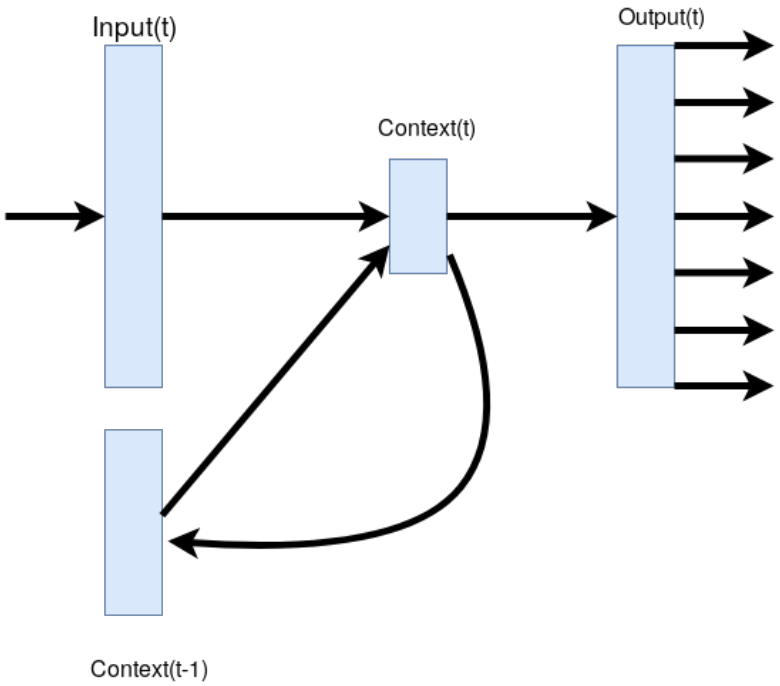

Fig. 3 RNNLM

This new language model showed 18% reduction of **Word-Error-Rate** in the WJS benchmark and 5% on NIST.

1.4 Markov Models

We often want to consider events that have an evolution over time, occur sequentially, and at each step the corresponding event depends on the preceding events in the sequence. This sequence of events is called a **Markov Chain** if has the the following properties called **Markov Properties**:

[1] RNNLM - Mikolov et al. (2009)

Given a sequence of events $X = (X_1, X_2, ..., X_T)$ that can assume values in $S = (s_1, s_2, ..., s_N)$

- Limited Horizon: The probability of an event depends on the previous recent events.

$$P(X_{t+1} = s_k | X_1 ... X_t) = P(X_{t+1} = s_k | X_t)$$

- Time Invariant (Stationary): The probability to see a variable do not depend on time itself.
- The definition of a Markov chain is given in terms of a *transition matrix A* and the *initial state probabilities* Π.

$$a_{ij} = P(X_{t+1} = s_j | X_t = s_i) \text{ and } \sum_j a_{ij} = 1$$

Definition

$$P(X_1 ... X_T) = P(X_1) * P(X_2|X_1) * P(X_3|X_1, X_2) * ... * P(X_T|X_1, ..., X_{T-1}) =$$
$$P(X_1) * P(X_2|X_1) * P(X_3|X_2) * ... * P(X_T|X_{T-1}) = \pi_{x_1} \Pi_{t=1}^{T-1} a x_1 x_{t+1}$$

Example: Given the following Markov chain compute **P(D, A, B)**

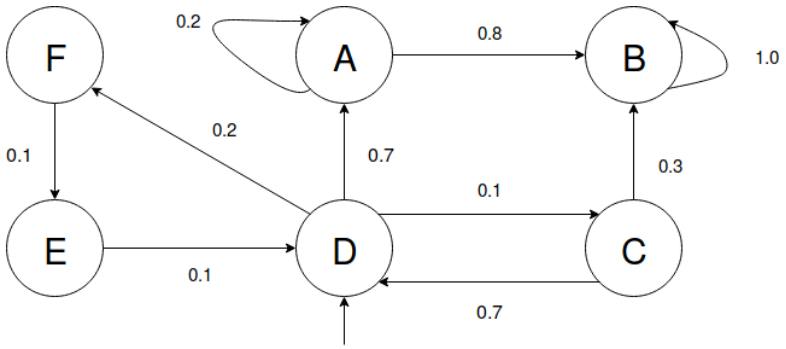

Fig. 4 Markov chain example

$P(D,A,B) = P(X_1 = D) * P(X_2 = A|X_1 = D) * P(X_3 = B|X_2 = A) = 1 * 0.7 * 0.8 = 0.056$

The Markov chain that we have seen is called a **Simple Markov Model** or **SMM**. In NLP we are interested on another type of Markov Model calle **Hidden Markov Model**[2] or simply **HMM**.

HMM are use vastly in NLP in many areas such as *Speech-Recognition*,*Part of Speech Tagging* and other.

Which are the motivations to study HMM?

[2] A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition - L. Rabiner (1989)

- We observe a sequence of symbols but we don't know the sequence of states that lead to the generation of those symbols.
- In the speech recognition the observation are the word and the hidden sequence are the part of speech that correspond to that words.

We can view HMM as an enriched version of SMM by stating that whenever a state is reached a symbol is emitted, so we have 2 sequences to consider now.

- $X = (X_1, ..., X_T)$ the sequence of the visited states.
- $O = (O_1, ..., O_T)$ the sequence of output symbols.

HMM is define by 5 elements:

- Set of states $S = s_1, ..., s_N$
- Set of output symbols $V = v_1, ..., v_M$
- Transition matrix $A = [a_{ij}]$
- Initial probability vector $\Pi[\pi_i]$
- Symbol emission matrix $B = [b_{ik}]$

$b_{ik} = P(O_t = v_k | X_t = s_i)$ and $\sum_k b_{ik} = 1$
is the probability to output v_k when the state is s_i

$$P(s_i, v_k) = P(X_t = s_i) * P(O_t = v_k | X_t = s_i)$$

We can use HMM for 3 main purposes:

1. Evaluation: Given the observation **O** and a HMM $\mu = (A, B, \Pi)$ compute $P(O|\mu)$. This is done using *Forward* procedures.
2. Recognition: Given the observation **O** and a HMM $\mu = (A, B, \Pi)$ how to chose the sequence of states that best explain the observations. In this case we use the *Viterbi* algorithm.
3. Training: Given the observation sequence **O** how do we adjust the model parameters of the HMM $\mu = (A, B, \Pi)$ to maximize the probability $P(O|\mu)$. *Forward/Backward (Baum/Welch)* algorithm is used.

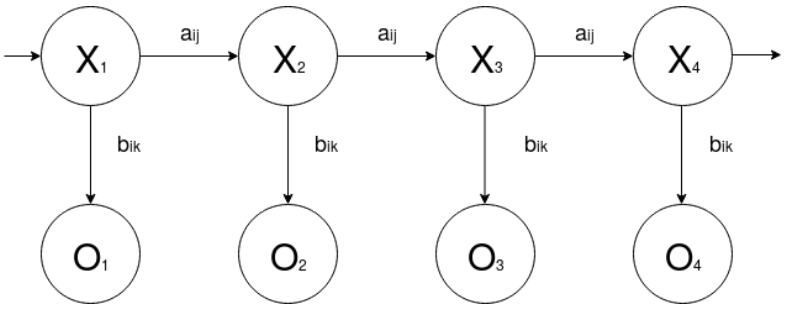

Fig. 5 Example of a HMM