INSTITUTO TECNOLÓGICO DE AERONÁUTICA

**Heládio Sampaio Lopes**

# NATURAL LANGUAGE PROCESSING FOR TREND FORECASTING

Final Paper
2020

## Course of Computer Engineering

CDU 621.38

**Heládio Sampaio Lopes**

# NATURAL LANGUAGE PROCESSING FOR TREND FORECASTING

Advisor

Prof. Dr. Filipe Alves Neto Verri (ITA)

**COMPUTER ENGINEERING**

São José dos Campos
Instituto Tecnológico de Aeronáutica

2020

**Cataloging-in Publication Data**
**Documentation and Information Division**

Lopes, Heládio Sampaio
Natural Language Processing for Trend Forecasting / Heládio Sampaio Lopes.
São José dos Campos, 2020.
25f.

Final paper (Undergraduation study) – Course of Computer Engineering– Instituto Tecnológico de Aeronáutica, 2020. Advisor: Prof. Dr. Filipe Alves Neto Verri.

1. Natural Language Processing. 2. Deep Learning. 3. Machine Learning. I. Instituto Tecnológico de Aeronáutica. II. Title.

## BIBLIOGRAPHIC REFERENCE

LOPES, Heládio Sampaio. **Natural Language Processing for Trend Forecasting**. 2020. 25f. Final paper (Undergraduation study) – Instituto Tecnológico de Aeronáutica, São José dos Campos.

## CESSION OF RIGHTS

AUTHOR'S NAME: Heládio Sampaio Lopes
PUBLICATION TITLE: Natural Language Processing for Trend Forecasting.
PUBLICATION KIND/YEAR: Final paper (Undergraduation study) / 2020



Heládio Sampaio Lopes
H8A St., 113
12228-460 – São José dos Campos–SP

# NATURAL LANGUAGE PROCESSING FOR TREND FORECASTING

This publication was accepted like Final Work of Undergraduation Study


Heládio Sampaio Lopes

Author

Filipe Alves Neto Verri (ITA)

Advisor

Inaldo Capistrano Costa

Course Coordinator of Computer Engineering



São José dos Campos: JUNE 19, 2020.

# Acknowledgments

Thank you

*"That's All Folks"*
— Looney Tunes,

# Resumo

Resumo

# Abstract

Abstract

# List of Figures

# List of Tables

# List of Abbreviations and Acronyms

| | |
|---|---|
| AI | Artificial Intelligence |
| BoW | Bag of Words |
| CBOW | Continuous Bag of Words |
| NLP | Natural Language Processing |
| ML | Machine Learning |
| TF-IDF | Term Frequency Inverse Document Frequency |
| IT | Information Technology |

# List of Symbols

# Contents

# 1 Introduction

## 1.1 Motivation

Every kind of expression, verbal or in writing, brings us a lot of information to be interpreted. Whether the topic is chosen, the tone used or the choice of words, everything can be interpreted, and then generate some useful information. Over the years, more and more knowledge is generated and we humans are not able to process such an amount of information. Natural language processing emerges as a technology capable of assisting us in this hard task.

Khurana *et al.* (2017) defines natural language processing, abbreviated by NLP, as a branch of artificial intelligence capable of making computers understand and extract information from human language. NLP can perform a lot of tasks, such as identifying different topics for a set of documents, classifying texts on predefined subjects, and beyond that extract the sentiment to know what people are saying about something.

## 1.2 Objective

Curious about the fast world's evolution, this works aims to explore and compare a several Natural Language Processing techniques to model the topic's evolution over time. With this in mind, evaluate te ability of those models to make predictions about future trends.

## 1.3 Organization of this work

# 2 Literature to Review

In this chapter, we will introduce the general concepts and techniques behind Natural Language Processing. We will cover all the necessary steps for extracting meaningful topics from texts.

## 2.1 Natural Language Processing

### 2.1.1 Text Processing Techniques

The key task to several machine learning problems consist in make a good data processing before applying any model. A clean data set can allow a model to increase its performance in the learning process, making a better identification in the patterns present in the variables. Hence, in the next sections, it will be discussed a few techniques to clear the text and prepare it for ML algorithms.

#### 2.1.1.1 Normalization

There is no right way to normalize text, this process has it is really important to put all text in the same level. A normalization process has a series of steps to be followed sequentially, all of then can be seen as 4 big tasks: stemming, lemmatization, stop words removal and everything else.

1. Stemming: Is the process of reduce inflected words to a primitive form, the stem. This method is able to remove the word's affixes to capture its base meaning, and still reducing the number of variations to save memory space. Figure 2.1 shows how some inflections for "connect" can be converted to its root form.

2. Lemmatization: similar to stemming, this step also reduce words to some primitive form, but with a little improvement. Lemmatization can returns the words to his dictionary form, based on its part of speech context. So it is possible to discriminate words with the same spelling but different meanings depending on the context.

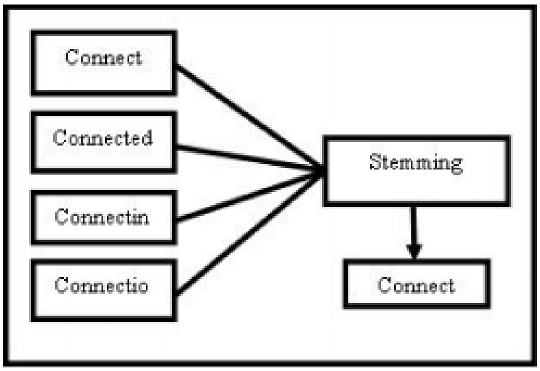



FIGURE 2.1 – Stemming process for “connect” variations (VIJAYARANI *et al.*, 2015).

3. Remove stop words: Many words can occurs a several times in a document without add any meaningful information, such as *the*, *is*, *at*, *which*, and *on*. Their high frequency can be seen as an obstacle to perform good results on NLP models, (KANNAN; GURUSAMY, 2014).

   There are some types to remove stop words, most of then based on evaluating the frequency of words in text, for more information see (**??**). But the classic and easier method is based on using a pre-compiled list of know words and removing then from text.

4. Everything else: Differently from the previous steps, the last one doesn't need any grammar rules or even a frequency analysis, it's purely text manipulation. It involves set all character to lowercase; remove numbers or convert then to word form; remove punctuation; expand contractions; convert special characters to ASCII form; and any other conversion needed.

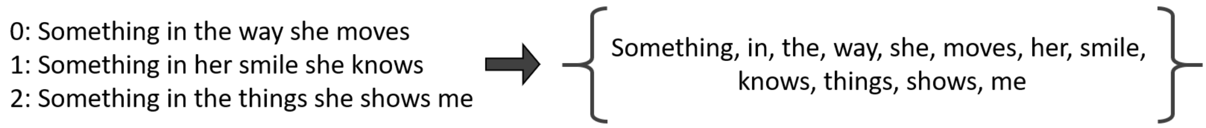
#### 2.1.1.2 Tokenization

Once the data is normalized, we need to know how to represent it. The tokenization process consists in splitting longer strings into meaningful small pieces called tokens. The most common way to tokenize a text is chunking it the into words, ie, given a piece of text the tokenize process will return a list of words.

#### 2.1.1.3 Bag of Words

The machine learning algorithms take numerical features as input, hence, it will bee necessary to represent the text in numerical form. With the Bag of Words model we can represent in matrix form a set of documents.

With the tokenization output we will have the lists representations for all documents in the data set. Those lists can be interpreted as vectors over the vector space of all unique tokens, also called by vocabulary. So, for a given sentence, we mark how many times its words appears in the list indexes where each entry corresponds to a word in the vocabulary. The Figure 2.2 show a simple example of how three sentences can be represented with BoW model.




| | Something | In | The | Way | She | Moves | Her | Smile | Knows | Things | Shows | Me |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| 2 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |

FIGURE 2.2 – Bag of Words example.

#### 2.1.1.4 TF-IDF

Term Frequency Inverse Document Frequency, TF-IDF for short, it is applied to a BoW to determine the relative frequency for words in a specific document when compared to the inverse proportion of that word over all documents in the collection. So, it can be determined how important are the words in a specific document.

From BoW, for the $i^{\text{th}}$ vocabulary's word in the $j^{\text{th}}$ document, its TF-IDF weight is:

$$w_{i,j} = \mathrm{tf}_{i,j} \times \log\left(\frac{N}{\mathrm{df}_i}\right). \tag{2.1}$$

Where, the term frequency, $\mathrm{tf}_{i,j}$, is how many time $i^{\text{th}}$ word appears in the $j^{\text{th}}$ document. The document frequency, $\mathrm{df}_i$, is the number of documents in which th $i_{\text{th}}$ vocabulary words is present. And, finally, $N$ is the size of the document collection, with a large number of documents this term can explodes, so the logarithmic function is applied to dampen this effect.

### 2.1.2 Word Embedding

The vectorization methods such as BoW and TF-IDF can be very useful, but they can not represent the words context. This means that the same words used in different contexts have the same representation, just as different words used with the same meaning are represented differently. Besides that, an one-hot encoding method, like BoW, presents a very sparse representation with high dimensionality.

The Word Embedding is a technique to represent words in vectors capable of capture the words context in a document. It is also able to smooth the high dimensionality effect by using much more compact vector to represent the words.

There are three most know way to perform a good word embedding. We will describe briefly each one of them below.

#### 2.1.2.1 Word Representations in Vector Space

The first great word embedding technique emerged when Google researchers proposed two architectures to build continuous vector representations of words. Word's context can be observed as the words that surround it in a sentence. Then using shallow neural networks, it is possible to calculate the word vector space based on word's context, (MIKOLOV *et al.*, 2013).

The first suggested approach is the continuous bag of words or CBOW, the left side of Figure 2.3 show its architecture. Here the neural networks is designed to predict, given the context, which word is most likely to appear. So, words with the same probability to appear can have a shared dimension in the words vector space.

The second approach is known by Skip-Gram, architecture at right in Figure 2.3. Very similar to CBOW, but instead of predicting the current word the Skip-Gram uses the current word as an input to a neural network to predict its context.

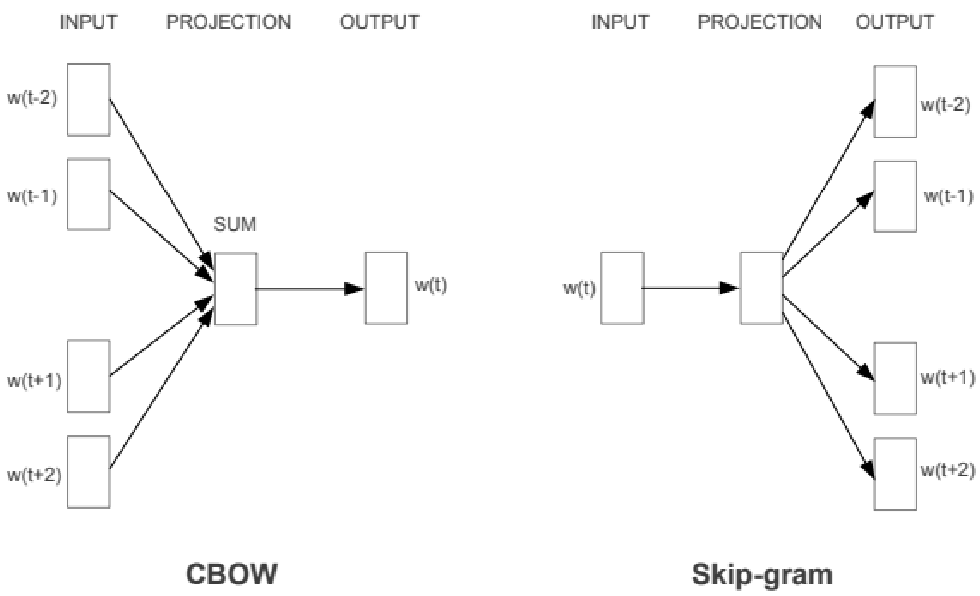



FIGURE 2.3 – Word2Vec architectures (MIKOLOV *et al.*, 2013).

After the network training process we can use the hidden layer weight matrix as an look up table to build the word embedding representation. The dimension for the vector space is managed by the number of neurons in the hidden layer.

#### 2.1.2.2 Global Vectors for Word Representation

Just a year later Pennington *et al.* (2014) arrives with a new approach to represent words in a vector space. The Global Vectors for Word Representation, or GloVe, method emerged by the need to consider some factors ignored by Skip-Gram.

Methods such as Skip-Gram learn their embedding by targeting words to their respective context, ignoring the fact that some words appear more in a context than others. Thus, this co-occurrence of words only adds more useless training examples, increasing the training complexity without adding relevant information.

GloVe, however, proposes to use the corpus statistics in a more efficient way. Using a weighted least squares model trained on a global word-word co-occurrence counts matrix. Thereby, it is possible to build a lookup table for the words in vocabulary and use it to represent them in a vector space.

#### 2.1.2.3 Word Vectors with Subword Information

Both Skip-Gram and GloVe provide a good vector representation for words, but there still are an unsolved problem, What to do with unknown words? To solve this question was proposed a new embedding technique which uses subword units to build a vector space, (BOJANOWSKI *et al.*, 2017).

Similar to Skip-Gram, this new method, the FastText, train its embedding by using a target to predict the context. However, instead of using the full words FastText goes a level deeper, breaking the words in $n$-grams, ie, the word becomes its own context. The Figure 2.4 shows how the word “apple” can be broken into $n$-grams.

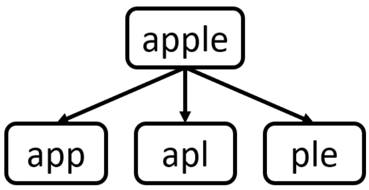



FIGURE 2.4 – Tri-gram representation for “apple” word.

There are a couple great advantages by using this method. It is now possible to generalize new words, or unseen in training data, since they have the same characters as known ones. Although it is possible to use available pre-trained models, the FastText requires less text to be trained, it can extract much more information from small pieces of text.

### 2.1.3 Topic Clustering

## 2.2 Machine Learning

# 3 Related Works

In the last chapter, we saw the theoretical foundation on NLP techniques. In this chapter, we will review in the literature some works that use the NLP techniques described to discover topics in a data set. In addition, we will show some applications for this type of task. And, finally, some final remarks to continue this work.

## 3.1 Topics Discovery

Finding meaningful topics in a document collection has been used for a lot of authors for the most various applications. For example, (HURTADO *et al.*, 2016) use topic modeling to inspecting research publications, patents, and technical reports aiming to model the evolution of the direction of research and forecast the near future trends in IT industry.

Using the titles and abstracts of a data set with over then six thousand academic papers between 2002 and 2010, mostly collected by Tang *et al.* (2008), they proposed a sentence-level association rule to discover the meaningful topics. After categorize the documents in topics they were capable of build time series for each found topic, marking how many times that topic was cited in a given year. So, they were able to build an ensemble of forecasters to study the patterns and relationships among topics over the years.

For a better understanding, the Figure 3.1 has a flowchart with their proposed framework for the topic discovery and forecasting.

This framework involves some well-known major steps of NLP processing. First, they convert the documents to their transactions form, ie, the phrases in each document will be considered individually during the process. Next, they perform the basics normalization steps which includes case conversion, tokenization, removing step words, part of speech tagging, stemming and lemmatization. It is also performed an additional step, specific to their application, removing verbs such as “exploiting”, “adapting” and “propose”, because they are very common in scientific publications and do not add much meaningful information.

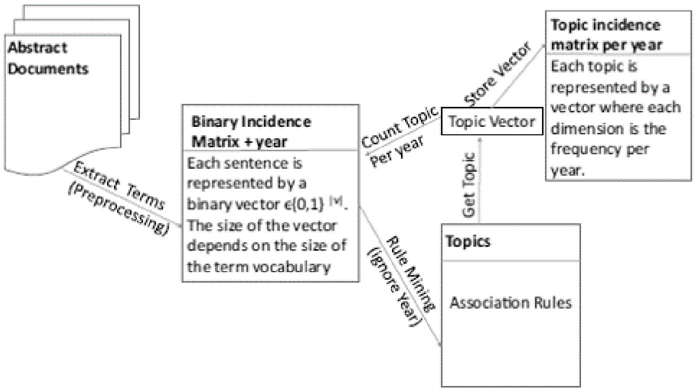



FIGURE 3.1 – Flowchart of the proposed framework (HURTADO *et al.*, 2016).

To vectorize the transactions, it is used a slight variation of BoW, Instead of word counting, it is only checked whether a word belongs to a transaction, this is called the binary incidence matrix. After these steps, comes the topic discovery part. Applying an association rule mining to the transactions and discovery their patterns. In order to avoid different topics with redundant words, is applied a rule refinement process that allows similar topics to be combined.

All documents present now belong to at least one identified topic in the set. It is time to create a topic incidence matrix that contains the count of times it is mentioned over the years. Finally, they make a ensemble forecasting to predict the future topic counting using the frameworks shown in Figure 3.2.

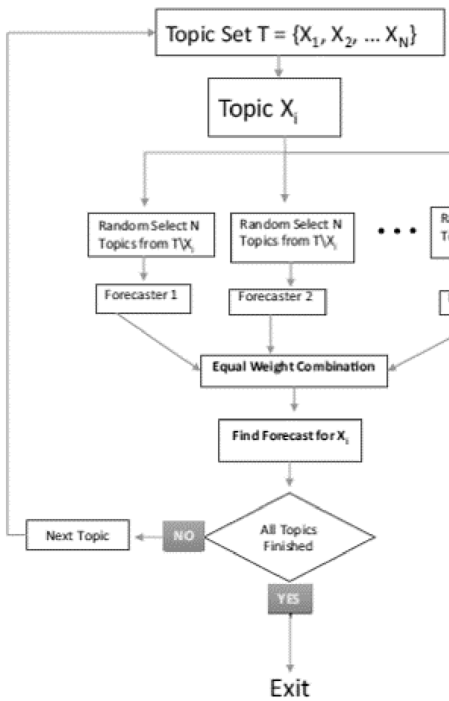



FIGURE 3.2 – Ensemble forecaster framework (HURTADO *et al.*, 2016).

## 3.2 Final Remarks

# 4 Roadmap

## 4.1 Hypothesis

## 4.2 Objective

## 4.3 Research method

## 4.4 Schedule

# Bibliography


BOJANOWSKI, P.; GRAVE, E.; JOULIN, A.; MIKOLOV, T. Enriching word vectors with subword information. **Transactions of the Association for Computational Linguistics**, v. 5, p. 135–146, 2017. ISSN 2307-387X.

HURTADO, J. L.; AGARWAL, A.; ZHU, X. Topic discovery and future trend forecasting for texts. **Journal of Big Data**, Springer, v. 3, n. 1, p. 7, 2016.

KANNAN, S.; GURUSAMY, V. Preprocessing techniques for text mining. **International Journal of Computer Science & Communication Networks**, v. 5, n. 1, p. 7–16, 2014.

KHURANA, D.; KOLI, A.; KHATTER, K.; SINGH, S. Natural language processing: State of the art, current trends and challenges. 08 2017.

MIKOLOV, T.; CHEN, K.; CORRADO, G.; DEAN, J. Efficient estimation of word representations in vector space. **arXiv preprint arXiv:1301.3781**, 2013.

PENNINGTON, J.; SOCHER, R.; MANNING, C. Glove: Global vectors for word representation. In: **Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)**. Doha, Qatar: Association for Computational Linguistics, 2014. p. 1532–1543. Disponível em: <https://www.aclweb.org/anthology/D14-1162>.

TANG, J.; ZHANG, J.; YAO, L.; LI, J.; ZHANG, L.; SU, Z. Arnetminer: extraction and mining of academic social networks. In: **Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining**. [S.l.: s.n.], 2008. p. 990–998.

VIJAYARANI, S.; ILAMATHI, M. J.; NITHYA, M. Preprocessing techniques for text mining-an overview. **International Journal of Computer Science & Communication Networks**, v. 5, n. 1, p. 7–16, 2015.

| FOLHA DE REGISTRO DO DOCUMENTO | | | |
|---|---|---|---|
| 1. CLASSIFICAÇÃO/TIPO<br>TC | 2. DATA<br>June 19th, 2020 | 3. DOCUMENTO N°<br>DCTA/ITA/DM-018/2015 | 4. N° DE PÁGINAS<br>25 |
| 5. TÍTULO E SUBTÍTULO:<br>Natural Language Processing for Trend Forecasting | | | |
| 6. AUTOR(ES):<br>**Heládio Sampaio Lopes** | | | |
| 7. INSTITUIÇÃO(ÕES)/ÓRGÃO(S) INTERNO(S)/DIVISÃO(ÕES):<br>Aeronautics Institute of Technology – ITA | | | |
| 8. PALAVRAS-CHAVE SUGERIDAS PELO AUTOR:<br>Natural Language Processing; Deep Learning; Machine Learning | | | |
| 9. PALAVRAS-CHAVE RESULTANTES DE INDEXAÇÃO:<br>Natural Language Processing; Deep Learning; Machine Learning | | | |
| 10. APRESENTAÇÃO: (**X**) **Nacional** ( ) **Internacional**<br>ITA, São José dos Campos, 2020. Trabalho de Graduação. 25 páginas. | | | |
| 11. RESUMO:<br>Resumo | | | |
| 12. GRAU DE SIGILO:<br>(**X**) **OSTENSIVO** ( ) **RESERVADO** ( ) **SECRETO** | | | |