Université Paris-Dauphine

Mathématiques et Informatique de la Décision et des Organisations

State of the art for Natural Language Processing

Authors:
Mayard Hippolyte
Randavel Olivier

Advisor:
Dessertine-panhard, Segolene

M2 Intelligence Artificielle Systèmes Données

April 2020

Contents

1 Introduction

Natural language processing (NLP) is a subfield of linguistics, computer science, information engineering, and artificial intelligence. It concerned with the interactions between computers and human languages, in particular how to program computers to process and analyze large amounts of natural language data. In Natural Language processing (NLP), there exist two main approaches. On the one hand, the Bag of Words models (BoW models) consider a document or a sentence as a bag containing words. These models focus on the words as well as their frequencies, regardless of the semantic relationships in the sentence. On the other hand, the Word Embeddings models associate words with respect to their meaning in a text. These models focus their learning on the context of a word appearance. From a text, Word Embeddings models give a distributed representation of words based on their usage. This allows words that are used in similar ways to results in having similar representations, naturally capturing their meaning.

Among the word embedding models, two methods are distinguished. The count-based methods (such as the Latent Semantic Analysis - LSA models) count the co-occurrences of a word with its neighbour words in a text corpus, while the predictive methods (based on Deep learning modelling like Word2Vec) try to directly predict a word from its neighbours in terms of learned small embedding vectors.

In this paper we will discuss several points. The first part will focus on data preparation as data must be normalized in order to find patterns between words. Then we will talk about statistical methods to extract keywords from text. The third part will consist in describing machine learning models. Finally an overview of tools developed by the GAFA will be presented.

2 Data preparation

Statistical or machine learning algorithms require normalized data to give a strong accuracy. In natural language processing, it is important to have clean and a large amount of data. Usually in NLP [1] it is not difficult to get not labeled data. Some tools such as beautiful soup, scrapy or sellenium help to collect data from website. The main issue is to process the data. The literature on this subject advice to simplify the data. With python functions such as re (regexp), uniquote, lower : it is possible to remove capital letters, emoticons and punctuation. Some more sophisticated algorithms are also used to reduce the amount of distinct words. Stemming word was introduced for the first time in 1968 by Julie Beth Lovins a computational linguistic. Then the Porter stemmer was published in 1980's by Martin Porter and finally scientists are using the snowball library. This nltk snowball library defines rules to keep only the root of a word by removing prefix and suffix. This word reduction can result in a not real word. Another method is named lemmatization. It is the algorithmic process of determining the lemma of a word based on its intended meaning. A plural word will be changed to its singular masculine form and a conjugate verb to its infinitive. In order to lemmatize a word, the literature advice to use spacy's library. All these methods reduce the number of distinct words from a document and remove irrelevant characters.

3 Statistical tools

This part presents several statistical methods commonly use when dealing with text data. These tools focus on all words and do not take into account any relationship between words. In fact plural and singular form of words represent two different words. Therefore, in order to improve these metrics it is important to prepare the data as shown in section 2.

3.1 TF-IDF

This metric helps to define keywords from document. It is call Term-Frequency - Inverse Document Frequency [2]. By normalizing the count, it returns only keywords that are important to the document. The formula was created in two times, TF was first presented by Hans Peter Luhn in 1957 and IDF by Karen Spärck Jones in 1972. This metric is built from TF and IDF formulas :

- The TF function is the Term Frequency that computes the frequency of word in a sentence. Let have n sentences in a document and m words. Then i represents a sentence and j a word. Finally you count the number of times the word j appears in a document i divide by the number of words in i

$$TF_{ij} = \sum_{l=1}^{k^{(i)}} \frac{1_{j=l}}{k^{(i)}}$$

- The IDF function is the Inverse Document Frequency that computes the importance of the word. It counts the number of documents divide by the number of times j does appear in a document. The logarithm is used to reduce the effect that can have a large document (big n) with a small j.

$$IDF_{ij} = \log \frac{n}{\sum_{l=1}^{n} 1_{l=j}}$$

Finally you have the TF-IDF function which the product of the two previous function.

$$x_{ij} = TF_{ij} * IDF_{ij} = \sum_{l=1}^{k^{(i)}} \frac{1_{j=l}}{k^{(i)}} * \log \frac{n}{\sum_{l=1}^{n} 1_{l=j}}$$

A way to improve the accuracy of TF-IDF is to reduce the number of distinct words in the document. The stematization or lemmatization methods can help. The TF-IDF metric is also used in different machine learning tools that we will describe in the next section 4. To implement TF-IDF it is possible to do it from scratch or generally people use the library *sklearn*.

3.2 Rapid Automatic Keyword Extraction : RAKE

The RAKE [3] method returns group of keywords. This algorithm uses punctuation and irrelevant words to construct group. A co-occurence matrix is built and displays the number of times that a word appear with another in a group. This metric is the the quotient of :

- $freq(w)$ the frequency of a word
- $deg(w)$ the degree of the word : the number of times a word appear plus the number of times it appears with another word in a group

The library commonly used is rake_nltk and was first introduced in 2015. It returns most relevant groups of words regarding several parameters such as the list of stopwords, the minimum and maximum length of group of words.

3.3 Topic modelling: LSA,LDA and PLDA

The general idea of topic modelling is that the semantics behind the analysed documents are actually governed by some hidden or latent variables that are not observed. Topic modelling aims at finding these hidden latent variables, that is, the topics.
Topic modelling, based on unsupervised text analytics algorithm, automatically discovers the hidden topics from the input documents, by building clusters of words that share the same semantic and contextual relationship.

In this section, we will focus on three different topic modelling methods: Latent Semantic Analysis (LSA), Probabilistic Latent Semantic Analysis (PLSA) and Latent Dirichlet Allocation (LDA).
These methods share 3 fundamental assumptions:

- Document have latent semantic structure (topics)
- Topic inference is based on word-document co-occurrences
- Words are related to topics, topics to documents

3.3.1 Latent Semantic Analysis (LSA)

The LSA topic modelling method [4] is based on bag of words models. From a document, this method compute the co-occurrence of words in a document. The resulting term-document matrix displays on the rows, the terms represented, and on the columns, the documents. LSA learns latent topics by performing a matrix decomposition on the document-term matrix using Singular Value Decomposition (SVD).
LSA is typically used as a dimension reduction or noise reducing technique. It typically replace raw counts in the document-term matrix with a TF-IDF score. Intuitively, a term has a large weight when it occurs frequently across the document, but infrequently across the corpus.

Advantages of LSA

- Easy to understand and implement
- Offers better results compared to the vector space models
- It is faster compared to other available algorithms because it involves document term matrix decomposition only.

Disadvantages of LSA

- Latent topic dimension depends upon the rank of the matrix. This means that the maximum number of topics will be, at maximum, equal to the rank of the matrix.
- LSA decomposed matrix is a highly dense matrix, so it is difficult to index individual dimension.
- There remain a doubt regarding the ability of this method to capture the multiple meanings and polysemy of a words.
- More (or as) difficult to implement than (as) LDA (Latent Dirichlet allocation).

3.3.2 Probabilistic Latent Semantic Analysis (PLSA)

The Probabilistic Latent Semantic Analysis methode [5] was introduced to overcome the disadvantages in LSA. PLSA automates document indexing, that is, based on a statistical latent class model for factor analysis of count data. This method improves LSA in a probabilistic sense by using a generative model. The main goal of PLSA is to identify and distinguish with multiple meanings and second, it discloses typical similarities by grouping together words that share a common context.

Advantages of PLSA

- Outperforms LSA method.
- Probabilistic model that can be easily extended and embedded in other more complicated models.

Disadvantages of PLSA

- Not a well-defined generative model : no way of generalizing to new, unseen documents
- Many free parameters (linear in the number of documents)
- The number of parameters grows linearly with the size of training documents, which leads to over-fitting.

3.3.3 Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation [6] is also a topic model that is used for discovering abstract topics from a collection of documents, a dimensionality reduction technique. LDA is a Bayesian method that uses dirichlet priors (sampling over a probability simplex or categorical distributions) for the document-topic and word-topic distributions, lending itself to better generalization than other methods. LDA extracts interpretable topics from a document corpus, where each topic is characterized by the words they are most strongly associated with. In comparing LSA with LDA, the latter has been found to be particularly suitable for documents containing multiple topics.

It contrasts with other approaches (for example, latent semantic indexing), as it is a generative probabilistic model, that is, a statistical model that allows the algorithm to generalize its approach to topic assignment to other, never-before-seen data points.

LDA is a particularly popular method for fitting a topic model. It treats each document as a mixture of topics, and each topic as a mixture of words. LDA is a mathematical method for estimating both these dimensions at the same time : finding the mixture of words that is associated with each topic, while also determining the mixture of topics that describes each document. This allows documents to "overlap" with each other in terms of content, rather than being separated into discrete groups, in a way that mirrors typical use of natural language.

Example
You have the following documents and assume you want to identify 2 topics
-I like bananas and oranges: 100% topic A
-Frogs and fish live in ponds: 100% topic B
-Kittens and puppies are fluffy: 100% topic B
-I had spinach and apple smoothie: 100% topic A

-My kittens love kale: 50% topic A and 50%topic B

Advantages of LDA

- LDA is a probabilistic model with interpretable topics.
- It allows for overlapping of words in different topics (i.e. You can get the same skill in different groups) and overlapping of topics (e. g. In different documents a two-topic model we could say "Document 1 is 90% topic A and 10% topic B, Document 2 is 30% topic A and 70% topic B and Document 3 is 100% topic A").
- mid-range of efficiency as far as ML algorithms: runtime could improve.

Disdvantages of LDA

- The number of topics is fixed and must be known ahead of time and it needs to set the number of topics manually. This can be time consuming and require several attempts before getting the final result.
- The topics have to be interpreted to decide whether they make sense
- Documents'length is a problem as it does not work well with small documents. The definition of documents is important.

4 Machine Learning tools

4.1 Word2Vec, GloVe and Fasttext

Word2vec was first introduced in 2013 by a team of researchers led by Tomas Mikolov at Google. This machine learning model is a two-layers neural networks that is trained to construct linguistic contexts of words. Word2Vec returns a large vector of hundreds of terms for each word of the corpus. These vectors correspond to the weights of the last hidden layer of the neural network. Using a distance formula such as cosinus similarity helps to see the similarity of worlds. Word2Vec takes into account a window size around the predicted word. This window represents the context of the word. It can be built using two different architectures. On the one hand the data-scientist can predict a word regarding his context this is named CBOW (Continous Bag of Words). On the other hand, the skip-gram method tries to predict context word regarding the center word.

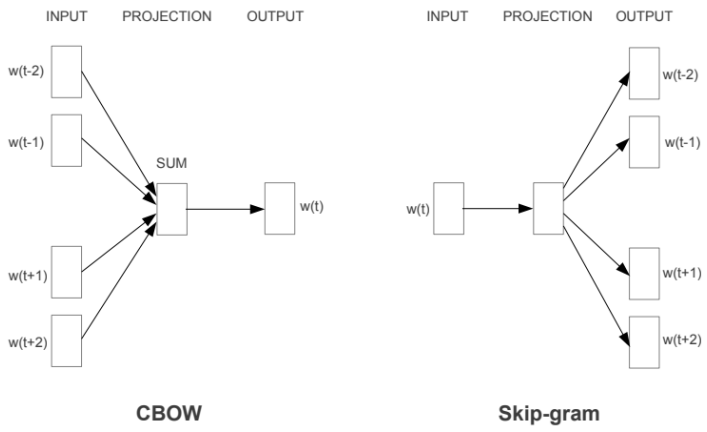

Figure 1: CBOW and Skip-gram structures

The literature [7] shows that CBOW is faster but skip-gram is more accurate. In order to evaluate the accuracy of these models two metrics are taken into account and are compared to a model that is considered correct : they define a comprehensive test set that contains five types of semantic questions, and nine types of syntactic questions. The semantic accuracy is concerned with matters such as sense, reference, presupposition and implication. The syntax accuracy is the study of the patterns of formation of sentences and phrases from words. Data-scientists are commonly using the gensim library developed by google, this model can be tuned using parameters such as window size, mathematical distance to compute similarity, architecture type or vectors size. Following the release of word2Vec, other similar machine learning models were developed. FastText was introduced in 2015 by Facebook. This model is created a window inside a word and try to predict a word. Each word is represented as a bag of character n-grams, so the overall word embedding is a sum of these character n-grams. Besides, we could also speak about GloVe an open-source project developed at Stanford in 2014. It is very similar to word2Vec but more efficient and faster as it computes the co-occurence matrix and doesn't require a window. Related works are available through these sources [7], [**glove**] and they show performance tables of the different models presented above.

4.2 LDA2Vec

LDA2VEC is an extension of both Word2Vec and LDA that jointly learns words, documents and topic vectors based on the skip-gram model of Word2Vec [8].
LDA2VEC builds on top of the skip-gram model of word2vec to generate word vectors. The interesting advantage of LDA2VEC relative to word2vec is that the former leverages a context vector to make the predictions. This context vector is created as the sum of two other vectors: the word vector (as in Word2Vec) and the document vector. The document vector is, itself, a weighted combination of two other components:

- the document weight vector, representing the"weights"(later to be transformed into percentages) of each topic in the document
- the topic matrix, representing each topic and its corresponding vector embedding

The power of LDA2vEC lies in the fact that it not only learns word embeddings (and context vector embeddings) for words, it simultaneously learns topic representations and document representation as well. Figure 2 below displays the structure of LDA2Vec.

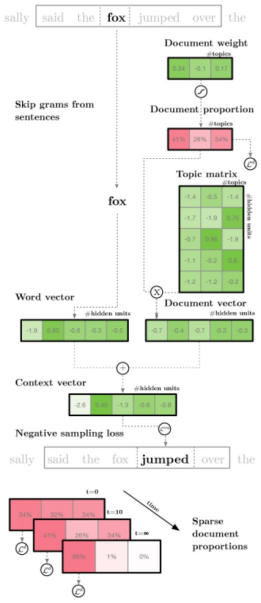

Figure 2: LDA2Vec structures

5 AWS-Google tools NLP.tex

This part will describe state of art models and ready to use tools created by the GAFA (Google, Amazon, Facebook and Apple). One of them is Amazon Comprehend [9]. This NLP tool was created by Amazon Web Services and is dedicating to people that doesn't have any experience with machine learning. This ready to use tool find insights and relationship in text. It helps company to filter spam emails, analyse product reviews. The tool automates all the process of data preparation and takes as input a raw text. As machine learning tools presented in the previous part, Amazon Comprehend is able to make syntax analysis by identifying adjectives and nouns within text, it also understands positive and negative customer reviews and help to summarize a document using its keyphrase extraction feature. Many companies trust in the power of these technology such as Roche a pharmaceuticals group or PWC group. An other model was recently released by Google in February 2020. This model is named T5 [10] for Text-To-Text Transfer Transformer and has been trained using C4 which state for Colossal Clean Crawled Corpus. This model is dedicated to data-scientists but a colab version helps the user to get started with the solution. This transfer learning model comes after BERT, ULMFIT or ELMo models that were released in 2018. The T5 model is a revolution in a way that the input and output are always text strings. In contrast to BERT-style models that can only output either a class label or a span of the input. The T5 text-to-text framework allows to use the same model, loss function, and hyperparameters on any NLP task, including machine translation, document summarization, question answering, and classification tasks. This work was accomplished by cleaning (deduplication, discarding incomplete sentences, and removing offensive or noisy content) and creating a large dataset (two times wikipedia dataset). This corpus is available through the tensorflow library. This state of art model was challenged with many other models and has the best results.

Table 1: Test set scores for T5 variants and previous results on the open-domain Natural Questions (NQ), WebQuestions (WQ), and TriviaQA (TQA) tasks.

	NQ	WQ	TQA
Chen et al. (2017)	–	20.7	–
Lee et al. (2019)	33.3	36.4	47.1
Min et al. (2019a)	28.1	–	50.9
Min et al. (2019b)	31.8	31.6	55.4
Asai et al. (2019)	32.6	–	–
Ling et al. (2020)	–	–	35.7
Guu et al. (2020)	40.4	40.7	–
Févry et al. (2020)	–	–	53.4
Karpukhin et al. (2020)	**41.5**	42.4	57.9
T5-Base	27.0	29.1	29.1
T5-Large	29.8	32.2	35.9
T5-3B	32.1	34.9	43.4
T5-11B	34.5	37.4	50.1
T5-11B + SSM	36.6	**44.7**	**60.5**

Finally on a more business side, Facebook has developed a tool to make automatic analyses on messenger. Companies can share customers messages from messenger app to detect keywords and make sentiment analyses. These tools is used to create BOT and to automate response to questions. Apple proposes a NLP toolbox to tokenize words, detect syntax and make word embedding.

6 Conclusion

The data-scientist's job is broad, it covers image recognition, prediction and natural language processing. Nowadays, data-scientists have accessed to trained algorithms and powerful computers to customize their model and answer to a business case. NLP is the easiest field to start data-science. Data is available everywhere and can be scrapped using powerful tools. But, it is always a challenge to create the most powerful model. New models are realising everywhere following competitions open to every researchers and students. Kaggle is offering a new challenge named the "NLP-2020 Chocolate Box Challenge".Besides challengedata from ENS school let people work on a judicial dataset : "NLP applied to judicial decisions parsing". NLP is a very active field of data-science. From all these competitions, hundreds of papers are written every year and improvements are made every days.

References

[1] Rachel Koenig. *NLP for Beginners: Cleaning Preprocessing Text Data.* URL: `https://towardsdatascience.com/nlp-for-beginners-cleaning-preprocessing-text-data-ae8e306bef0f`. (accessed: 09.03.2020).

[2] Juan Ramos. *Using TF-IDF to Determine Word Relevance in Document Queries.* URL: `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.121.1424&rep=rep1&type=pdf`. (accessed: 06.03.2020).

[3] Nick Cramer Stuart Rose Dave Engel and Wendy Cowley. *Automatic keyword extraction from individual documents.* URL: `https://www.researchgate.net/publication/227988510_Automatic_Keyword_Extraction_from_Individual_Documents`. (accessed: 03.03.2020).

[4] S. Deerwester et al. "Indexing by latent semantic analysis." In: *Journal of the American Society for Information Science 41* (1990), pp. 391–407.

[5] Thomas Hofmann. "Learning the Similarity of Documents: An Information-Geometric Approach to Document Retrieval and Categorization". In: *Advances in Neural Information Processing Systems 12*. Ed. by S. A. Solla, T. K. Leen, and K. Müller. MIT Press, 2000, pp. 914–920. URL: http://papers.nips.cc/paper/1654-learning-the-similarity-of-documents-an-information-geometric-approach-to-document-retrieval-and-categorization.pdf.

[6] David Blei, Andrew Ng, and Michael Jordan. "Latent Dirichlet Allocation". In: vol. 3. Jan. 2001, pp. 601–608.

[7] Tomas Mikolov et al. *Efficient Estimation of Word Representations in Vector Space*. 2013. arXiv: 1301.3781 [cs.CL].

[8] Christopher E. Moody. "Mixing Dirichlet Topic Models and Word Embeddings to Make lda2vec". In: *CoRR* abs/1605.02019 (2016). arXiv: 1605.02019. URL: http://arxiv.org/abs/1605.02019.

[9] AWS. *Amazon Comprehend*. URL: https://aws.amazon.com/comprehend/. (accessed: 21.05.2020).

[10] Adam Roberts Adam Roberts. *Exploring Transfer Learning with T5: the Text-To-Text Transfer Transformer*. URL: https://ai.googleblog.com/2020/02/exploring-transfer-learning-with-t5.html. (accessed: 21.05.2020).