Automatic detection of persuasion attempts on social networks

Ruben Teimas

Universidade de Évora, Évora, Portugal
May 24, 2022
m47753@alunos.uevora.pt

Abstract: This paper makes a revision on multi-label text classification, more specifically for detection of persuasion techniques and hate speech on social networks. In it we talk about text pre-processing, word embeddings, problem transformation and (in a lightly way) more robust approaches like transformers. We also provide some real word usage examples for the techniques described above.

Keywords: NLP · Deep-Learning · Multi-label Classification · Machine-Learning

1 Introduction

With the appearance of *Web 2.0* regular internet users were able to upload their own content to the web by using platforms such as blogs and social networks, amongst them: *Facebook*, *Twitter*, *Instagram* and *Reddit*. Throughout the years time spent by users on those platforms has increased. It is reported that, on average, each American spent more than 1300 hours on social networks during the year of 2021. [9]

Whilst allowing users to post as they are pleased increases the diversity of content which, at first sight, is good for free speech it also comes with the risk of having undesired content. People that post undesired content are often referred as *trolls*, usually users with anonymous identities who use techniques such as *name calling*, *whataboutism*, presenting irrelevant data (*red herring*), amongst others.

Troll's existence can be perceived as a minor inconvenient or even innocuous, however they can play a big part on real world events. An well known case is the possible interference of Russian *trolls* on the *USA*'s 2016 Presidential Elections. [7]

SemEval has released multiple challenges toward detection of persuasion attempts based on text.

2 Persuasion Attempts

Detection of persuasion attempts can be performed using binary classification: either the content incorporates persuasion attempts, or it doesn't. *Facebook* re-

leased a binary classification challenge [6] for detecting Hate Speech on multimodal memes.

However, the previous approach tells us very little about the persuasion techniques itself and how the content tries to persuade the agent. For that reason, instead of simply detecting persuasion attempts, many challenges ask participants not only to detect them but also specify which techniques are being used.

For most of the *SemEval* challenges the list of techniques used is the one defined on a paper called *Fine-Grained Analysis of Propaganda in News Article* [1]. In it there are defined 18 techniques (classes) and they can all appear at the same time for one entry which faces us with a multi-label classification problem.

3 Text pre-processing

A large part of data from social networks are is based on natural language (also know as *text*).

However, natural language involves a lot of complexity: there are nouns, verbs (with different verb forms), pronouns... We might also encounter some slang or simply some meaningless words, also know as stopwords (like *the*, *and*, *this*).

The process of dealing with all this problems is known as *text normalization* and can sometimes be something very similar to the following diagram.

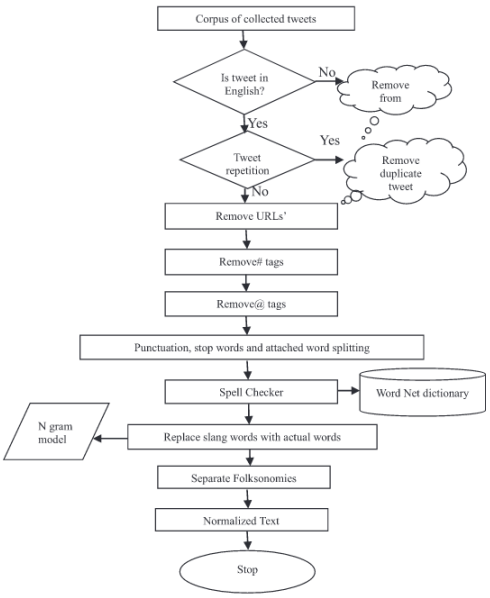

Fig. 1. Text normalization diagram proposed. [8]

After having normalized our text, we still need a way for the algorithms to be fed with the text, however computers cannot understand natural language, for that reason normalized text is converted into numerical vectors. There are some conventional approaches like Bag of Words (BoW) and Term Frequency-Inverse Document Frequency ($Tf\text{-}Idf$) and some newer approaches like *word2vec*.

3.1 Bag of Words

Given the number of different words on the dataset, each word is given an index of the vector and its value is number of occurrences on each entry.

It is a simple way of turning textual elements into numerical vectors, however it does have some disadvantages, amongst them:

- Vectors get very large as the number of words increase.
- Vectors tend to be sparse, as there are a lot of 0's.
- We are retaining no information on the grammar of the sentences nor on the ordering of the words in the text.

3.2 Tf-Idf

It consists of 2 terms: *Term frequency* and *Inverse document frequency*. While Tf is relative to the number of occurrences of the word the Idf gives a notion of the importance of each word. Higher importance is given to less frequent words.

Comporative to BoW:

- It contains information on the more important words and the less important ones as well.
- It usually performs better in machine learning models.

3.3 word2vec

It is a more sophisticate way of obtaining the word embeddings, mainly because not only it encodes the words but its representation also contains some of the word features which makes it easier to identify correlations and proximity between words.

There are two main architectures which yield the success of word2vec. The skip-gram and CBoW architectures.

Continous Bag of Words: This architecture is very similar to a feed forward neural network. This model architecture essentially tries to predict a target word from a list of context words. The intuition behind this model is quite simple: given a phrase "Have a great day" , we will choose our target word to be "a" and our context words to be ["have", "great", "day"]. What this model will do is take the distributed representations of the context words to try and predict the target word.

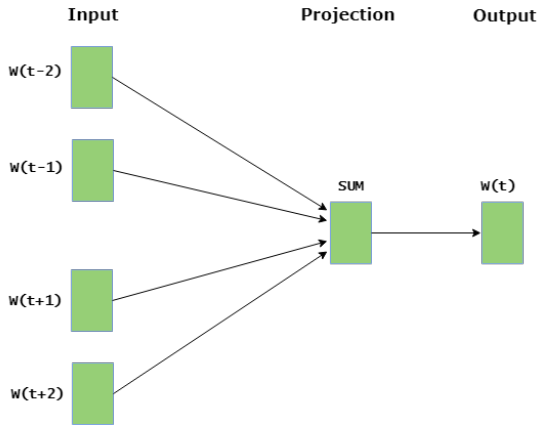

Fig. 2. CBoW architecture. [11]

Skip-gram: The skip-gram model is a simple neural network with one hidden layer trained in order to predict the probability of a given word being present when an input word is present.

You can think of it as the opposite to the *CBoW* as it takes the current word as an input and tries to accurately predict the words before and after this current word.

This architecture can be visually represented as described on *Fig. 2*:

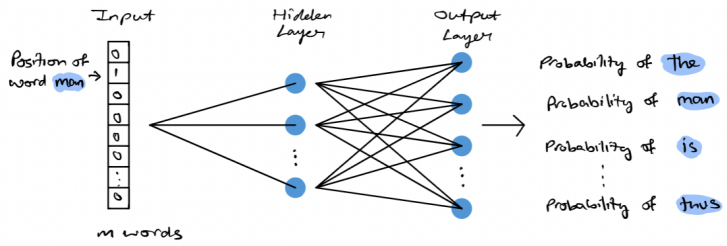

Fig. 3. Skip-gram architecture. [11]

A corpus can be represented as a vector of size N, where each element in N corresponds to a word in the corpus. During the training process, we have a pair of target and context words, the input array will have 0 in all elements except for the target word. The target word will be equal to 1. The hidden layer will learn the embedding representation of each word, yielding a d-dimensional embedding space. The output layer is a dense layer with a softmax activation function. The output layer will essentially yield a vector of the same size as the input, each element in the vector will consist of a probability. This probability

indicates the similarity between the target word and the associated word in the corpus. [11]

4 Multi-label classification

Before talking about multi-label classification it is important to differentiate it from multi-class. When presented with a multi-class classification problem there are multiple labels but each instance of the dataset can only be classified as one of those classes. On multi-label classification problem multiple an instance from the dataset can be classified with multiple classes at the same time.

Taking a more conventional approach (*machine-learning* techniques) can be quite complicated since most of the supervised learning tasks have been carried out using single label classification and solved as binary or multi-class classification problems.

These restrictions lead researchers to usually follow different paths like the ones presented on the images bellow.

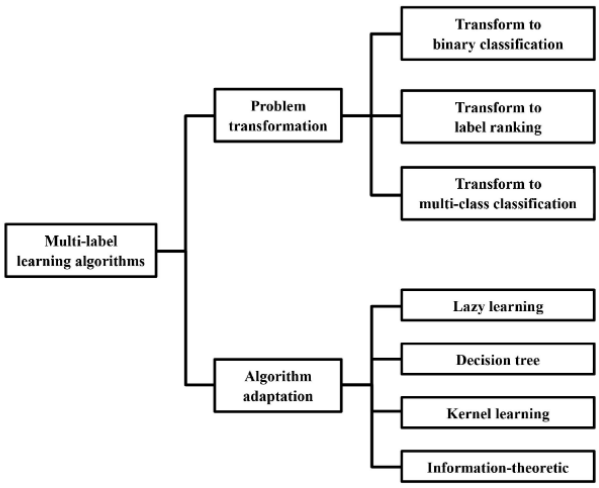

Fig. 4. Categorization of representative multi-label approaches. [13]

5 Problem Transformation

This approach consists of turning a multi-label classification (MLC) problem into a binary or multi-class classification problems. Some of the most well known techniques are *Binary Relevance* and *Label Powerset*.

5.1 Binary Relevance

The basic idea of this approach is to decompose the multi-label learning problem into q independent binary classification problems, where each binary classification problem corresponds to a possible label in the label space.

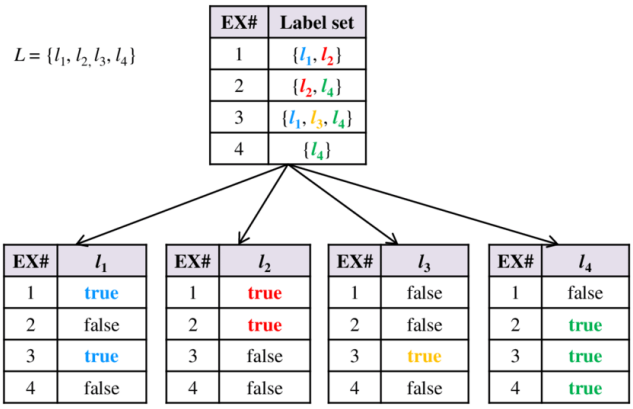

Fig. 5. Binary relevance illustration. [12]

The result of each instance is the combination of all the classifiers. This approach does take relationship between classes in account as it treats them independently, so in order for it to be a good choice we must try to make sure that the classes are disjointed.

5.2 Label Powerset

The approache's name comes from the fact that it considers each member of the power set of labels in the training set as a single label.

By doing so, it takes into account the possible relations between classes (which is an improvement over the *Binary Relevance*) and treats each group of labels as a new class. It also allows to make label ranking for each class, which orders labels by its relevance.

Nonetheless, Label powerset does have some drawback, amongst them:

- **Computational complexity**: If the number of labels is very large, the number of combinations will also increase, which will lead to the need of performing a lot of computational tasks.
- **Can't predict unseen labels**: If a combination of labels has never been used as a result, the classifier cannot predict it, this causes overfitting.

Example	x_1	...	x_7	Adventure	Drama	Comedy	Class
Game of Thrones	x_{11}	...	x_{17}	1	1	0	C110
The Big Bang Theory	x_{21}	...	x_{27}	0	0	1	C001
Rick and Morty	x_{31}	...	x_{37}	1	0	1	C101
College Romance	x_{41}	...	x_{47}	0	1	1	C011

Fig. 6. Label powerset illustration.

5.3 Real world usage

In 2019 was published a paper[4] which discusses multi-label text classification for abusive language and hate speech detection including detecting the target, category, and level of hate speech in Indonesian Twitter using machine learning approach with Support Vector Machine (SVM), Naive Bayes (NB), and Random Forest Decision Tree (RFDT) classifier and Binary Relevance (BR), Label Powerset (LP).

The problem only had 3 classes which makes it a good candidate for problem transformation. The researches did not only tried different models but also different ways to pre-process the text and feature extraction. In the end, the best results came out of word unigram (feature extraction), Random Forest using Label powerset as the transformation method with an accuracy of 77.36%.

On the following year, the same researches release a new paper [5] with some improvements, in it they proposed to also use *part of speech* (also known as grammatical tagging), emojis and the *word2vec* for word embeddings. They managed to get a minor upgrade to the previous accuracy getting 79.85%.

However, this result was not obtained using *word2vec*, which slightly worse results. This corpus used was relatively small and for that reason *word2vec* was not able to train the model properly

6 Deep-Learning on *MLC* problems

Even tho more conventional approaches can give good results, in the past few years more sophisticated Neural Networks (*NN*) have appeared. They are more robust than the previous solutions and can usually perform more efficient computation which means they're faster to train (on the same amount of data). Amongst them there is the LSTM architecture.

6.1 LSTM

It stands for *Long Short Term Memory* and they are an improvement over the *RNN*. Recurrent Neural Networks had been used to perform text classification,

however they had a lot of difficulties retrieving information from very long past moments.

LSTM overcomes this problem by adding an internal state which will also be used in the output calculation. It also hads a forget and recall gates which dictates which information is kept and which information is discarded.

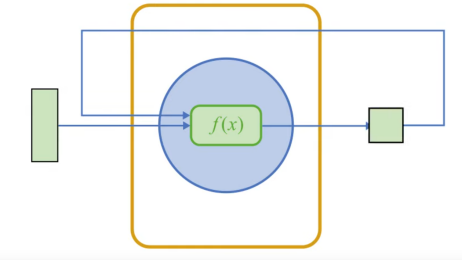

Fig. 7. RNN architecture.

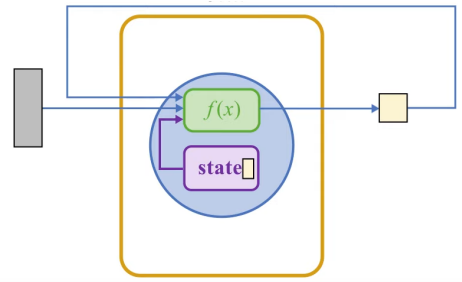

Fig. 8. LSTM architecture.

For some time *LSTM* stood as the go-to architecture for text classification, however that was about to change when the paper "*Attention is all you need*" [10] was released, and so *Transformers* where introduced.

6.2 Transformers

Transformers (like BERT, AlBERT or RoBERTa) are now the go-to to multi-label text-classification. Most of the models come already pre-trained requiring the researcher to "only" fine-tuning them. But first, we need to understand what a transformer is.

At its core, it contains a stack of Encoder layers and Decoder layers. The Encoder stack and the Decoder stack each have their corresponding Embedding

layers for their respective inputs. Finally, there is an Output layer to generate the final output.

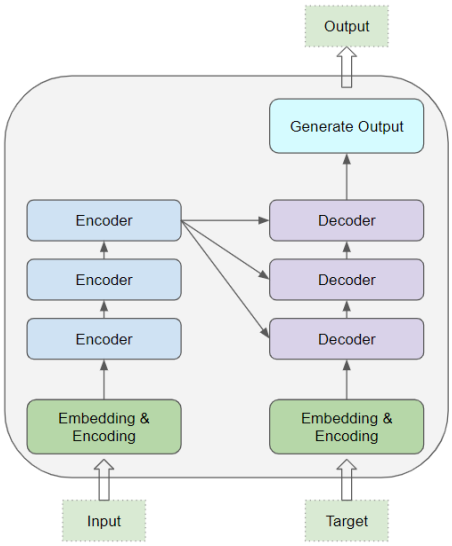

Fig. 9. Over the top Transformer architecture [3].

The Encoder contains the Self-attention layer that computes the relationship between different words in the sequence, as well as a Feed-forward layer. The Decoder, besides the 2 same components as the Encoder, has an Encoder-Decoder attention layer inbetween. But what does *attention* really does? While processing a word, *attention* enables the model to focus on other words in the input that are closely related to that word. Lets have 2 examples:

- The cat drank the milk because **it** was hungry.
- The cat drank the milk because **it** was sweet.

In the first sentence, the word 'it' refers to 'cat', while in the second it refers to 'milk. When the model processes the word 'it', self-attention gives the model more information about its meaning so that it can associate 'it' with the correct word.

6.3 Real world usage

One of the latest *SemEval* challenges focus on "Detection of Persuasion Techniques in Texts and Images" with 1 of the sub-tasks being text related only. After it was finished, a paper [2] was released describing the best models presented as well as it results.

This problem was way more complex than the one presented before, it used 20 labels (whilst the previous used 3) and it probably wouldn't perform well using a more conventional approach.

On the paper released most of the submissions revolved around transformers, with the best one having a *F1-micro* score of .593. The winning group explored pre-trained models *PTMs* with *Focal Loss* (which assigns higher weights to sparse samples) due to data imbalance.

7 Conclusion

By doing a revision on multi-label text classification we realise that even though it has many applications, for example, *detection of persuastion techniques on social networks*, it is a very though problems with a vast number of techniques and approaches to explore and although some architectures are better in theory (like Transformers) there is still the need to made a lot of tests and experiments.

Bibliography

[1] Giovanni Da San Martino, Seunghak Yu, Alberto Barrón-Cedeño, Rostislav Petrov, and Preslav Nakov. Fine-grained analysis of propaganda in news article. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5636–5646, Hong Kong, China, November 2019. Association for Computational Linguistics.

[2] Dimitar Dimitrov, Bishr Bin Ali, Shaden Shaar, Firoj Alam, Fabrizio Silvestri, Hamed Firooz, Preslav Nakov, and Giovanni Da San Martino. Semeval-2021 task 6: Detection of persuasion techniques in texts and images, 2021.

[3] Ketan Doshi. Transformers explained visually (part 1): Overview of functionality, Jun 2021.

[4] Muhammad Okky Ibrohim and Indra Budi. Multi-label hate speech and abusive language detection in Indonesian Twitter. In *Proceedings of the Third Workshop on Abusive Language Online*, pages 46–57, Florence, Italy, August 2019. Association for Computational Linguistics.

[5] Muhammad Okky Ibrohim, Muhammad Akbar Setiadi, and Indra Budi. Identification of hate speech and abusive language on indonesian twitter using the word2vec, part of speech and emoji features. In *Proceedings of the International Conference on Advanced Information Science and System*, AISS '19, New York, NY, USA, 2019. Association for Computing Machinery.

[6] Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. The hateful memes challenge: Detecting hate speech in multimodal memes. *CoRR*, abs/2005.04790, 2020.

[7] Katharine Schwab. We analyzed 1.8 million images on twitter to learn how russian trolls operate, Aug 2020.

[8] Tajinder Singh and Madhu Kumari. Role of text pre-processing in twitter sentiment analysis. *Procedia Computer Science*, 89:549–554, 2016. Twelfth International Conference on Communication Networks, ICCN 2016, August 19– 21, 2016, Bangalore, India Twelfth International Conference on Data Mining and Warehousing, ICDMW 2016, August 19-21, 2016, Bangalore, India Twelfth International Conference on Image and Signal Processing, ICISP 2016, August 19-21, 2016, Bangalore, India.

[9] Peter Suciu. Americans spent on average more than 1,300 hours on social media last year, Dec 2021.

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[11] Vatsal. Word2vec explained, May 2022.

[12] Vedrana Vidulin. Searching for credible relations in machine learning. 02 2012.

[13] Min-Ling Zhang and Zhi-Hua Zhou. A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 26(8):1819–1837, 2014.