Handle Multilinguality in Text classification

Nesma Mahmoud: B8771
Mahmoud Shoush: B87770

Natural language processing (LTAT.01.001) - Spring 2019

Abstract

Text classification is a supervised machine learning method used to classify sentences or text documents into one or more defined categories. Its a widely used natural language processing task playing an important role in Spam filtering, sentiment analysis, categorization of news articles and many other business related issues. Handling multilinguality in text classification is a crucial process, and there is no standard approach to handle it. In this project we propose and compare between Four different methods to tackle the multilinguality in text classification (MLTC):

- **Multiple monolingual classification approach.**
- **Comparing joint multilingual approach.**
- **Joint translated monolingual.**
- **Round trip translation.**

We did experiments using different models, and find the round trip translation is the best one to handle MLTC. **Still need more explanation**.

Input Text
Classification Model
Output Category (Sports)

Figure 1: Text Classification

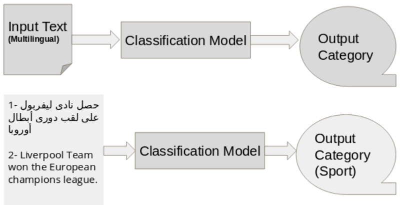

Input Text (Multilingual)
Classification Model
Output Category
1- حصل نادى ليفربول على لقب دورى أبطال أوروبا
2- Liverpool Team won the European champions league.
Classification Model
Output Category (Sport)

Figure 2: Multilingual Text Classification.

1 Introduction

Natural Language Processing (NLP) is the technology used to teach computers to understand and generate the humans natural language. NLP is the driving force behind abundance set of applications such as, language translation applications like Watson IBM, Word Processors such as Microsoft Word and Grammarly that employ NLP to check grammatical accuracy of texts, and Personal assistant applications such as OK Google, Siri, Cortana, and Alexa etc. There are many approaches to automatic text classification, which can be grouped into three different types of systems, Rule-based systems, Machine Learning based systems, and Hybrid systems.

Text classification is one of the major tasks in NLP (a.k.a. text categorization or text tagging) is the task of assigning a set of predefined categories to free-text. Text classification can be used to organize, structure, and categorize pretty much anything. For example, new articles can be organized by topics, support tickets can be organized by urgency, chat conversations can be organized by language, brand mentions can be organized by sentiment, and so on. For example (see figure 3), imagine you have a text with different categories European Champions league is a quite hard and tough for many teams. A classifier can take this text as an input, analyze its content, and then automatically assign relevant category, such as Sport and tag as difficult.

Multilingual text classification (MLTC) (see figure 4) means collections of documents written in several languages with different set of labels that we need to classify and predict [1].

2 Related Work

There has been research on MLTC in the case of enough training documents being available for every language. In [1] they proposed a method that combines different monolingual classifiers in order to get a new classifier as good as the best monolingual one which has the ability to deliver all the best performance measures (precision, recall and F1) possible. In [2] they provide linguistic arguments against existing approaches and devise a novel solution that exploits background knowledge from ontologies and lexical resources. In [3] They proposed a method which is language-independent because it does not depend on the low of grammar by using Character N-gram. Therefore, they can classify multi-language into some categories using only one program. In [4] This paper presents a novel approach to Cross-lingual Distillation for Text Classification that builds on model distillation, which adapts and extends a framework originally proposed for model compression. Using soft probabilistic predictions for the documents in a label-rich language as the (induced) supervisory labels in a parallel corpus of documents, we train classifiers successfully for new languages in which labeled training data are not available.

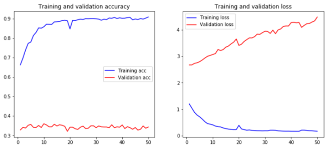


Figure 3: Text classification with preprocessing

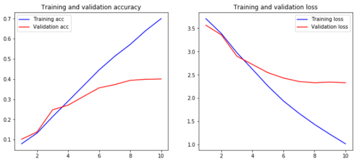


Figure 4: Text Classification without preprocessing.

3 Proposed Approach

In this section, we will explain in details our proposed approach through the whole four main stages that we mentioned before. The proposed methods have been implemented based on the STOA NLP frameworks such as Bert [5], Elmo [6], and Alennlp [7]. The code and detailed implementation can be found in Handling_Multilinguality repo. For this project we find a multilingual-text-categorization-dataset that contains 33 languages and 45 categories. After exploring the dataset we discovered that it's not balanced (some of the languages are more bigger than the others), and for this project we used only five languages (English, Arabic, French, Spanish, and Estonian).

3.1 Text Preprocessing

In this section, we will talk about the basic steps of text preprocessing. These steps are needed for transferring text from human language to machine-readable format for further processing. After a text is obtained, we start with text normalization. Text normalization includes converting all letters to lower case, remove stop words, numbers, punctuation, blank space, and URL, stemming, and lemmatization. This dataset contains blog posts labeled by language and by category.

In this step we used a Keras framework to build our classifier and apply it on EnglishData only with/without preprocessing, (see figure 3) and (4) respectively. We can see obviously that heavy preprocessing gives negative impact on the classification process. Here can check the implementation of with preprocessing, and without pre-processing.

3.2 Multiple Monolingual Classification

This is the first method to tackle the the MLTC problem. In this approach we classify all of the languages (English, Arabic, French, Spanish, and Estonian) by a separate classification model (see figure 5) using different models such as Flair, and Allennlp. Here you can see check the Allennlp implementation for each language separately. Also we used Flair Embedding and Glove over English dataset with 2 categories as a POC due to resources shortage here and when it compares to Allennlp implementation we got accuracy around 65 % which is an improvement over Allennlp. From this step we find that using Flair has an advantage over Allennlp implementation.

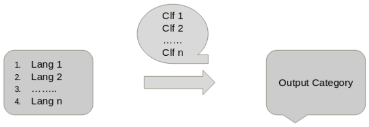


Figure 5: Multiple Monolingual Classification

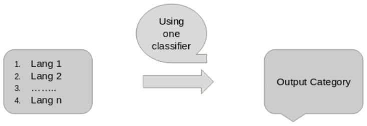


Figure 6: Joint multilingual approach.

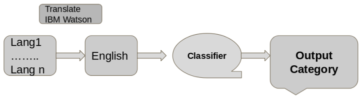


Figure 7: Joint translated monolingual

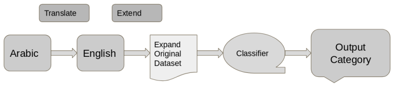


Figure 8: Round Trip Translation.

3.3 Joint multilingual approach

In the second method that we propose to handle MLTC, we classify all of the languages together with single classification system (see figure 6) using different models as the previous step. here you cand find the Allennlp implementation for this approach, however in this step all languages are concatenated and the dataset splited into 80%, 20% for training step and 20% from all languages was selected as validation. Also, we did validation using a subset from the English dataset, which was excluded before spliting the whole data frame into train and dev and got accuracy around 75%.

3.4 Joint translated monolingual

This is the third method that we propose to handle the MLTC problem. In this method all languages are translated into one super-language - English - and then classified all together (see figure 7). For this approach we tried to use different public translation service such as googletrans, and Watson IBM langugue translator, but in this public services the free usage is so limited per character and per account credentials then accordingly we didn't able to apply this step for the whole dataset however we minimize the number of words per row to fifty words. You can find detailed implementation here. **Add results about accuracy??????**

3.5 Round Trip Translation.

In this last approach we used round trip translation to expand the available data. In this method we translated all languages to English, then add the translated part to the original dataset to expand it, after that apply our classifier to the expanded dataset(see figure 9). Here you can find the implementation of the RT applied Arabic **Add other languages** and we got accuracy around 24%.

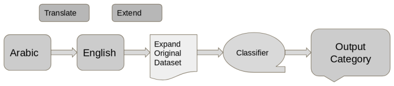
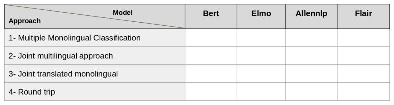


Figure 9: Round Trip Approach



Model Approach	Bert	Elmo	Allennlp	Flair
1- Multiple Monolingual Classification				
2- Joint multilingual approach				
3- Joint translated monolingual				
4- Round trip				

Figure 10: Results conclusion.

4 Conclusion and discussion

In this section we will conclude our experiments and results. We proposed four approaches to tackle the multilinguality in text classification for each approach we used different models frameworks. **Add explanation after fulfilling the missing results**

5 Future Work

This project just a start in MLTC using different models, and there is alot to do to enhance the process of text classification with multilingual languages. In this section I will try to include some ideas to enhance this project. First we need to build a complete dataset for the purpose of MLTC since the available datasets are not balanced and limited to specifics topics and objectives. Also we need to use fee-based service for translation between different languages at the same time to solve the problem of accounts usage.

References

[1] Gonalves, T.,& Quaresma, P. (2010). Multilingual text classification through combination of monolingual classifiers. In Proceedings of the 4th Workshop on Legal Ontologies and Artificial Intelligence Techniques (pp. 29-38).

[2] De Melo, G. and Siersdorfer, S., 2007, April. Multilingual text classification using ontologies. In European Conference on Information Retrieval (pp. 541-548). Springer, Berlin, Heidelberg.

[3] Suzuki, Makoto, Naohide Yamagishi, Yi-Ching Tsai, and Shigeichi Hirasawa. ”Multilingual text categorization using Character N-gram.” In 2008 IEEE Conference on Soft Computing in Industrial Applications, pp. 49-54. IEEE, 2008.

[4] Xu, Ruochen, and Yiming Yang. ”Cross-lingual distillation for text classification.” arXiv preprint arXiv:1705.02073 (2017).

[5] Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. ”Bert: Pretraining of deep bidirectional transformers for language understanding.” arXiv preprint arXiv:1810.04805 (2018).

[6] Peters, Matthew E., Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. ”Deep contextualized word representations.” arXiv preprint arXiv:1802.05365 (2018).

[7] Gardner, Matt, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. ”AllenNLP: A deep semantic natural language processing platform.” arXiv preprint arXiv:1803.07640 (2018).