Genre classification using Balanced Winnow in the DEFT 2014 challenge

Eva D'hondt
LIMSI–CNRS, Rue John von Neumann, 91405 Orsay
eva.dhondt@limsi.fr

Résumé. Dans ce rapport, nous présentons le travail effectué sur la première tâche du challenge DEFT 2014. Cette édition portait sur la classification de genre pour des textes littéraires français. Dans notre approche, nous avons développé trois types de caractéristiques : des mots lemmatisés, des caractéristiques stylometric et des caractéristiques intègrant une certaine forme de connaissance du monde. Nos expériences de classification ont été effectuées à l'aide de l'algorithme de classification 'Balanced Winnow'. Les meilleurs résultats ont été obtenus par la combinaison des trois types de caractéristiques.

Abstract. In this report we present the work done on the first subtask of the DEFT 2014 challenge which dealt with genre classification of French literary texts. In our approach we developed three types of features : lemmatized words, stylometric features and features that incorporate some form of world knowledge. Subsequent classification experiments were performed using the Balanced Winnow classifier. We submitted three different runs of which the best-scoring one combined all features.

Mots-clés : catégorisation de text, DEFT, genre littéraire.

Keywords: text classification, DEFT, literary genre.

1 Introduction

DEFT (Défi Fouille de Textes) is a yearly competition which focuses on text mining of French texts. Each year the DEFT organisers present multiple text mining tasks within a different domain. This year's challenge focused on the processing and mining of French literary and scientific texts. Like last year the organisers developed 4 separate tasks. In this article we report our participation in the first task : genre classification of literary short stories and poems.

Genre classification is a task which focuses on both the *content* and the *structure* of the given texts : While some literary genres have fixed and recurring themes – for example, a police novel will often contain words such as 'crime', 'victim', 'chase', ... – other genres such as poetry employ less fixed theme sets. To correctly recognise these literary genres, features must be devised that capture differences in the distribution of punctuation and other style elements between the different genres.

Over the last forty years there has been considerable work done on genre classification. (Kessler *et al.*, 1997; Stamatatos *et al.*, 2000; Finn & Kushmerick, 2006) However, such studies often use a fairly broad definition of genre, and consequently need to differentiate between very different types of texts from both written and spoken language, prose and poetry, ... The task presented in this DEFT challenge is more fine-grained as it concerns different categories of written texts within the literary domain. Moreover, in this DEFT challenge, documents can belong to multiple categories rendering it a multilabel, multicategory classification task.

2 Corpus Description

The data of this year's track was furnished by Short Edition, a printing house specializing in 'short literature' such as short stories and poetry. The training data provided consists of 2328 documents in XML format. Each document has been manually labeled by the editors at Short Edition and contains at least 1 and at most 5 labels. The majority of the training documents (64%) contain 2 subcategory labels.

The classification scheme is organised as follows : Each document belongs to one or more subcategories which themselves fall within sections. The main category ('type') of document is given as part of the training information. A document can only be of one type : Either 'très très court', 'poème' or 'nouvelles'.

In total there are 7 sections ('poésie','autres','émotions','chronique','noir','jeunesse','fantastique-sf') and a total of 45 subcategories (here ordered per section) :

– Poésie : alexandrins, chanson, haikus, slam, sonnets, vers libres, prose, comptine et fable ;
– Chronique : arts, gastronomie, histoire, nature, sciences - médical, société, spiritualité, sport, voyage ;
– Emotions : amitié, colère, drame, enfance, erotisme, famille, humour, instant de vie, mélancolie - solitude, nostalgie - souvenirs, romance ;
– Fantastique-SF : ésotérique, fantasy, merveilleux, science-fiction, surnaturel ;
– Jeunesse : jeunesse ;
– Noir : aventure, horreur, policier, suspens, thriller ;
– Autres : allégorie, conte, fantaisiste, lettre, autre.

As a subcategory can only fall within one section these dependencies could be used to build hierarchical classifiers. This is left to future work however ; The experiments reported in this paper only deal with 'flat' classification, i.e. classification between the different subcategories.

There is a clear imbalance in size between the different subcategories : Both the 'instant de vie' and 'vers libres' labels occur more then 1000 times in the training corpus, and up to a third of all the training documents carry 'société', 'romance' or 'drame' labels.

The 995 test documents were only released during the testing phase of the competition. After the track was closed we look at the label distribution within this set and found it to be similar to that of the training corpus.

3 Data processing and feature generation

We extracted the text from the title, content and type fields in the original XML documents. The content field contained formatting information as well as free text which enabled us to calculate statistics on line length and other formatting choices (see infra). We created and extracted three different types of features :

3.1 Bag of Words

The title and text from the content field were tokenized and lemmatized using Treetagger for French. Where Treetagger was not able to provide a lemma, the original word form was used. Over all, Treetagger proved very successful : 92% of words in the training corpus were lemmatised. The output of the lemmatisation process was then used as bag-of-words features in the subsequent experiments. Please note that we include the type information of the document as part of the bag-of-words features in the experiments.

3.2 Stylometric features

Since the original formatting information was available in the form of XML tags for line breaks, paragraphs, ... , we were able to calculate the following stylometric statistics :

– average sentence length (in number of words) ;
– average line length (in number of words) ;
– average paragraph length (in number of words) ;
– number of paragraphs ;
– average number of punctuation marks used per sentence.

All measures were binned and used as separate features in training experiments. [1] Experiments on the training set showed that especially the metrics on average line length and number of paragraphs prove useful, mostly in distinguishing between the more structured poems and longer running texts.

1. All training experiments were conducted using 5-fold cross-validation on the entire training corpus.

3.3 World knowledge features

We calculated three additional features that used some form of world knowledge to add information to the text :

Number of emotive words in the document We counted the number of emotion words using a self-defined list of French words denoting emotion. [2] This count was then normalized against the total number of words in the document and binned. Training experiments showed that this feature was useful to distinguish subcategories of the 'émotion' section from other subcategories.

Number of stopwords in the document Using the french stopword list from the NLTK package we calculated a similar metric to the one reported above. This metric did not prove informative during training experiments.

Part-of-Speech tags Next to the lemmatized words we allowed the PoS tags derived by Treetagger as to be used as classification features. During training experiments we found a slight improvement by only adding NOUN, VERB and ADJECTIVE tags. [3]

4 Balanced Winnow algorithm

All experiments were performed using the Balanced Winnow algorithm as implemented in the Linguistic Classification System (Koster *et al.*, 2001), hereafter referred to as LCS. Balanced Winnow is one of the lesser known classification algorithms. It is akin to the Perceptron algorithm. During training it learns two weights (positive and negative) for each feature t per subcategory c. The difference between the positive and negative weight is the effective (Winnow) weight of a feature. During the training phase, the Winnow algorithm assigns labels to training documents. If the document is assigned a correct label, the feature weights are not changed. If the document is assigned an erroneous label, the positive weights for the active features, that lead to the mistake, will be demoted, while their negative weights are promoted. If a document is not assigned the correct subcategory label, the positive weights for the active features in that subcategory will be promoted while the negative weights are demoted, thus making it more likely to arrive at the correct classification in the next training iteration. At testing time, the sum of the Winnow weights of the active features for the test document determine the Winnow score per subcategory. The Winnow algorithm has been used in multiple text classification task and proven particularly succesful in classification tasks with a larger number of features (D'hondt *et al.*, 2013).

5 Submitted runs

For the official evaluation we submitted three runs with the following three configurations :

Run 1 Bag-of-Words features only ;
Run 2 Bag-of-Words + average line length + number of paragraphs + stopword ratio + emotion ratio ;
Run 3 Bag-of-Words + average line length + number of paragraphs + stopword ratio + emotion ratio + PoS features (only nouns, verbs and adjectives).

For each run we optimized the LCS parameters on a held-out set of the training data. We also configured the LCS to return at least 1 and most 5 labels per test document in the output but only if these subcategory labels had a Winnow weight higher than a cut-off rate of 0.8. We used the order in which subcategory labels were returned by the classifiers to determine the rankings in the submitted runs.

The runs were evaluated by the track organisers using the normalized discounted cumulative gain (nDCG) measure. This metric measures the performance of a system based on the graded relevance of the returned subcategory labels where a higher-ranked relevant label has a greater impact on the ultimate score then a lower-ranked relevant label.

To give an idea of the difficulty of the task we have included two other traditional measures from Information Retrieval : Precision and Recall.

The submitted runs resulted in the following scores :

2. Based on the list found at `https://fbcdn-sphotos-e-a.akamaihd.net/hphotos-ak-prn1/16290_539895756057646_1977252385_n.png`
3. For greater generality we converted the Treetagger tags to their generic categories, e.g. 'VER :futu' to 'VERB'.

Runs	nDCG of submitted run	Precision	Recall
Run 1	0.3817	0.4493	0.3116
Run 2	0.3800	0.4359	0.3248
Run 3	0.3900	0.4619	0.3216

TABLE 1 – Results of submitted runs to genre classification subtask.

We did not see any significant improvement from either the stylometric features or the added PoS features, although the latter lead to the best result. Compared to the other two participants in the track we achieved the lowest scores. The highest-scoring submitted run in the competition achieved a nDCG of 0.5248.

In preparation for this report we performed a post-run analysis of Run 3 output and subcategory models to gain a better understanding why the classification results are low. We found several contributing factors : First, the classification output for the run showed that on average 1.6 labels per document were returned. This indicates that our cut-off was set too high which caused many potentially relevant subcategory labels to be dismissed. Furthermore, selecting only a subset of the output to be evaluated is good practice when trying to attain high Precision scores but detrimental to nDCG scores : A relevant document at a low rank will still contribute to the overall cumulative score, while an irrelevant document does not have a negative impact. Therefore, to improve nDCG scores it is better to evaluate on the full rankings, rather than a subset. We consequently reconfigured the LCS to output full rankings, i.e. for each document it returns scores for all 45 subcategories, and reran our experiments. This resulted in much better nDCG scores which even surpassed the highest official score :

Runs	nDCG of submitted run	nDGC of rerun
Run 1	0.3817	0.6311
Run 2	0.3800	0.6233
Run 3	0.3900	0.6333

TABLE 2 – Comparison of submitted runs and scores of reruns.

Please note that these experiments used the same models as in the official runs : We did not retrain the classifiers but only changed the output configuration of the LCS. The Precision and Recall scores remain the same as reported in Table 5.

Second, we found that the Precision and Recall scores differed greatly between subcategories. Subcategories with a lot of training material like 'instant de vie' and 'vers libres' which together make up about half of the training material, have large, well balanced models while smaller categories like 'haikus' have too little training material to construct adequate models. A further complicating factor is the fact that the corpus is multilabel : Most of the documents that carry the label of an infrequent subcategory are also labeled for one of the larger subcategories. This affects training as it means that the document can no longer be used as negative training material to distinguish the smaller subcategory from its larger counterpart.

Close examination of the classification models [4] shows that the constructed stylometric features prove informative for some categories. For example, the high ratio of stopwords in a document proved a determining feature to distinguish haikus from the rest of the corpus. However, the impact of these features is limited to only a handful of subcategory models. We suspect that our binning method is too crude and valuable information is lost in mapping the calculated ratios to nominal features. As the LCS can only classify using nominal features, future experiments will be be conducted using another classification method.

Références

D'HONDT E., VERBERNE S., KOSTER C. & BOVES L. (2013). Text Representations for Patent Classification. *Computational Linguistics*, **39**(3), 755–775.

FINN A. & KUSHMERICK N. (2006). Learning to classify documents according to genre. *Journal of The American Society for Information Science and Technology*, **57**, 1506–1518.

4. LCS produces human-readable lists of features which associated Winnow weights.

KESSLER B., NUMBERG G. & SCHÜTZE H. (1997). Automatic detection of text genre. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, p. 32–38 : Association for Computational Linguistics.

KOSTER C., SEUTTER M. & BENEY J. (2001). Classifying patent applications with winnow. In *Proceedings Benelearn 2001*, p. 19–26, Antwerpen.

STAMATATOS E., FAKOTAKIS N. & KOKKINAKIS G. (2000). Automatic text categorization in terms of genre and author. *Computational linguistics*, **26**(4), 471–495.