Experiments with Clustering-based Features for Sentence Classification in Medical Publications: Macquarie Test's participation in the ALTA 2012 shared task.

Diego Mollá
Department of Computing
Macquarie University
Sydney, NSW 2109, Australia
diego.molla-aliod@mq.edu.au

Abstract

In our contribution to the ALTA 2012 shared task we experimented with the use of cluster-based features for sentence classification. In a first stage we cluster the documents according to the distribution of sentence labels. We then use this information as a feature in standard classifiers. We observed that the cluster-based feature improved the results for Naive-Bayes classifiers but not for better-informed classifiers such as MaxEnt or Logistic Regression.

1 Introduction

In this paper we describe the experiments that led to our participation to the ALTA 2012 shared task. The ALTA shared tasks[1] are programming competitions where all participants attempt to solve a problem based on the same data. The participants are given annotated sample data that can be used to develop their systems, and unannotated test data that is used to submit the results of their runs. There are no constraints about what techniques of information are used to produce the final results, other than that the process should be fully automatic.

The 2012 task was about classifying sentences of medical publications according to the PIBOSO taxonomy. PIBOSO (Kim et al., 2011) is an alternative to PICO for the specification of the main types of information useful for evidence-based medicine. The taxonomy specifies the following types: **P**opulation, **I**ntervention, **B**ackground, **O**utcome, **S**tudy design, and **O**ther. The dataset was provided by NICTA[2] and consisted of 1,000 medical abstracts extracted from PubMed split into an annotated training set of 800 abstracts and an unannotated test set of 200 abstracts. The competition was hosted by "Kaggle in Class"[3].

Each sentence of each abstract can have multiple labels, one per sentence type. The "other" label is special in that it applies only to sentences that cannot be categorised into any of the other categories. The "other" label is therefore disjoint from the other labels. Every sentence has at least one label.

2 Approach

The task can be approached as a multi-label sequence classification problem. As a sequence classification problem, one can attempt to train a sequence classifier such as Conditional Random Fields (CRF), as was done by Kim et al. (2011). As a multi-label classification problem, one can attempt to train multiple binary classifiers, one per target label. We followed the latter approach.

It has been observed that the abstracts of different publication types present different characteristics that can be exploited. This lead Sarker and Mollá (2010) to the implementation simple but effective rule-based classifiers that determine some of the key publication types for evidence based medicine. In our contribution to the ALTA shared task, we want to use information about different publication types to determine the actual sentence labels of the abstract.

To recover the publication types one can attempt to use the meta-data available in PubMed. However, as mentioned by Sarker and Mollá (2010), only a percentage of the PubMed abstracts is annotated with the publication type. Also,

[1] http://alta.asn.au/events/sharedtask2012/

[2] http://www.nicta.com.au/

[3] http://inclass.kaggle.com/c/alta-nicta-challenge2

Diego Mollá. 2012. Experiments with Clustering-based Features for Sentence Classification in Medical Publications: Macquarie Test's participation in the ALTA 2012 shared task. In *Proceedings of Australasian Language Technology Association Workshop*, pages 139−142.

time limitations did not let us attempt to recover the PubMed information before the competition deadline. Alternatively, one can attempt to use a classifier to determine the abstract type, as done by Sarker and Mollá (2010).

Our approach was based on a third option. We use the sentence distribution present in the abstract to determine the abstract type. In other words, we frame the task of determining the abstract type as a task of clustering. We attempt to determine natural clusters of abstracts according to the actual sentence distributions in the abstracts, and then use this information to determine the labels of the abstract sentences.

Our approach runs into a chicken-and-egg problem: to cluster the abstracts we need to know the distribution of their sentence labels. But to determine the sentence labels we need to know the cluster to which the abstract belongs. To break this cycle we use the following procedure:

At the training stage:

1. Use the annotated data to train a set of classifiers (one per target label) to determine a first guess of the sentence labels.
2. Replace the annotated information with the information predicted by these classifiers, and cluster the abstracts according to the distribution of predicted sentence labels (more on this below).
3. Train a new set of classifiers to determine the final prediction of the sentence labels. The classifier features include, among other features, information about the cluster ID of the abstract to which the sentence belongs.

Then, at the prediction stage:

1. Use the first set of classifiers to obtain a first guess of the sentence labels.
2. Use the clusters calculated during the training stage to determine the cluster ID of the abstracts of the test set.
3. Feed the cluster ID to the second set of classifiers to obtain the final sentence type prediction.

2.1 Clustering the abstracts

The clustering phase clusters the abstracts according the distribution of sentence labels. In particular, each abstract is represented as a vector, where each vector element represents the relative frequency of a sentence label. For example, if abstract A contains 10 sentences such that there are 2 with label "background", 1 with label "population", 2 with label "study design", 3 with label "intervention", 3 with label "outcome", and 1 with label "other", then A is represented as $(0.2, 0.1, 0.2, 0.3, 0.3, 0.2, 0.1)$. Note that a sentence may have several labels, so the sum of all features of the vector is greater than or equal to 1.

We use K-means to cluster the abstracts. We then use the cluster centroid information to determine the cluster ID of unseen abstracts at the prediction stage. In particular, at prediction type an abstract is assigned the cluster ID whose centroid is closest according to the clustering algorithm inherent distance measure.

In preliminary experiments we divided the abstracts into different zones and computed the label distributions in each zone. The rationale is that different parts of the abstract are expected to feature different label distributions. For example, the beginning of the abstract would have a relatively larger proportion of "background" sentences, and the end would have a relatively larger proportion of "outcome" sentences. However, our preliminary experiments did not show significant differences in the results with respect to the number of zones. Therefore, in the final experiments we used the complete sentence distribution of the as one unique zone, as described at the beginning of this section.

Our preliminary experiments gave best results for a cluster size of $K = 4$ and we used that number in the final experiments. We initially used NLTK's implementation of K-Means and submitted our results to Kaggle using this implementation. However, in subsequent experiments we replaced NLTK's implementation with our own implementation because NLTK's implementation was not stable and would often crash, especially for values of $K >= 4$. In our final implementation of K-Means we run 100 instances of the cluster algorithm with different initialisation values and choose the run with lower final cost. The chosen distance measure is $\sum_i (x_i - c_i)^2$, where x_i is feature i of the abstract, and c_i is feature i of the centroid of the cluster candidate.

3 Results

For the initial experiments we used NLTK's Naive Bayes classifiers. We experimented with the following features:

p Sentence position in the abstract.

np Normalised sentence position. The position is normalised by dividing the value of p with the total number of sentences in the abstract.

w Word unigrams.

s Stem unigrams.

c Cluster ID as returned by the clustering algorithm.

The results of the intial experiments are shown in Table 1. Rows in the table indicate the first classifier, and columns indicate the second classifier. Thus, the best results (in boldface) are obtained with a first set of classifiers that use word unigrams plus the normalised sentence position, and a second set of classifiers that use the cluster information and the normalised sentence position.

Due to time constraints we were not able to try all combinations of features, but we can observe that the cluster information generally improves the $F1$ scores. We can also observe that the word information is not very useful, presumably because the correlation between some of the features degrades the performance of the Naive Bayes classifiers.

In the second round of experiments we used NLTK's MaxEnt classifier. We decided to use MaxEnt because it handles correlated features and therefore better results are expected. As Table 1 shows, the results are considerably better. Now, word unigram features are decidedly better, but the impact of the cluster information is reduced. MaxEnt with cluster information is only marginally better than the run without cluster information, and in fact the difference was not greater than the variation of values that were produced among repeated runs of the algorithms.

We performed very few experiments with the MaxEnt classifier because of a practical problem: shortly after running the experiments and submitting to Kaggle, NLTK's MaxEnt classifier stopped working. We attributed this to an upgrade of our system to a newer release of Ubuntu, which presumably carried a less stable version of NLTK. We subsequently implemented a Logistic Regression classifier from scratch and carried a few further experiments. The most relevant ones are included in Table 1. We only tested the impact using all features due to time constraints, and to the presumption that using only sentence positions would likely produce results very similar to those of the Naive Bayes classifiers, as was observed with the MaxEnt method.

The Logistic Regression classifier used a simple gradient descent optimisation algorithm. Due to time constraints, however, we forced it to stop after 50 iterations. We observed that the runs that did not use the cluster information reached closer to convergence than those that used the cluster information, and we attribute to this the fact that the runs with cluster information had slightly worse $F1$. Overall the results were slightly worse than with NLTK's MaxEnt classifiers, presumably due to the fact that the optimisation algorithm was stopped before convergence.

The value in boldface in the MaxEnt component of Table 1 shows the best result. This corresponds to a first and second set of classifiers that use all the available features. This set up of classifiers was used for the run submitted to Kaggle which achieved best results, with an AUC of 0.943. That placed us in third position in the overall ranking.

Table 2 shows the results of several of the runs submitted to Kaggle. Note that, whereas in Table 1 we used a partition of 70% of the training set for training and 30% for testing, in Table 2 we used the complete training set for training and the unannotated test set for the submission to Kaggle. Note also that Kaggle used AUC as the evaluation measure. Column *prob* shows the results when we submitted class probabilities. Column *theshold* shows the results when we submitted labels 0 and 1 according to the classifier threshold. We observe the expected degradation of results due to the ties. Overall, $F1$ and AUC *(prob)* preserved the same order, but AUC *(threshold)* presented discrepancies, again presumably because of the presence of ties.

4 Summary and Conclusions

We tested the use of cluster-based features for the prediction of sentence labels of medical abstracts. We used multiple binary classifiers, one per sentence label, in two stages. The first stage used

With Naive Bayes classifiers

	$-$	$c+p$	$c+np$	$c+w$	$c+w+np$	$c+s+np$
p	0.440	0.572				
np	0.555		0.577			
w	0.448		0.610	0.442		
$w+np$	0.471		**0.611**		0.468	
$s+np$						0.485

With MaxEnt classifiers

	$-$	$c+p$	$c+np$	$c+w$	$c+w+np$	$c+s+np$
p						
np			0.574			
w		0.646		0.704		
$w+np$	0.740				**0.759**	
$ws+np$						0.758

With Logistic Regression classifiers

	$-$	$c+p$	$c+np$	$c+w$	$c+w+np$	$c+s+np$
$w+np$	**0.757**				0.747	

Table 1: $F1$ scores with a Naive Bayes classifiers.

	$F1$	*AUC (prob)*	*AUC (threshold)*
MaxEnt $w+np-c+w+np$	0.759	0.943	
NB $w-c+np$	0.610	0.896	
NB $np-c+np$	0.577	0.888	
NB $p-c+p$	0.572	0.873	0.673
NB w	0.448		0.727
NB $w-c+w$	0.442	0.793	
NB p	0.440		0.654

Table 2: Comparison between $F1$ in our results and AUC in the results submitted to Kaggle.

standard features, and the second stage incorporated cluster-based information.

We observed that, whereas cluster-based information improved results in Naive Bayes classifiers, it did not improve results in better informed classifiers such as MaxEnt or Logistic Regression. Time constraints did not allow us to perform comprehensive tests, but it appears that cluster-based information as derived in this study is not sufficiently informative. So, after all, a simple set of features based on word unigrams and sentence positions fed to multiple MaxEnt or Logistic Regression classifiers were enough to obtain reasonably good results for this task.

Further work on this line includes the incorporation of additional features at the clustering stage. It is also worth testing the impact of publication types as annotated by MetaMap or as generated by Sarker and Mollá (2010).

References

Su Nam Kim, David Martinez, Lawrence Cavedon, and Lars Yencken. 2011. Automatic classification of sentences to support Evidence Based Medicine. *BMC bioinformatics*, 12 Suppl 2:S5, January.

Abeed Sarker and Diego Mollá. 2010. A Rule-based Approach for Automatic Identification of Publication Types of Medical Papers. In *Proceedings of the Fifteenth Australasian Document Computing Symposium.*