An Evaluation of Multilabel Classification for the Automatic Annotation of Texts

Eneldo Loza Mencía
Knowledge Engineering Group
Technische Universität Darmstadt
eneldo@ke.tu-darmstadt.de

Abstract

This article presents the formulation of an information extraction as a multilabel classification problem. This representation allows for exploiting annotation overlappings and correlations. The standard multiclass approach is compared to different multilabel classification algorithms.

1 Introduction

In recent years, more and more approaches have appeared that translate the IE task into a classical classification problem, which is nowadays considered the most popular approach. The most common approach is to transform each text position, i.e. usually each text token in the document, into a classification example. This is often called boundary classification or sequential tagging/labeling. The class information of the instance depends on whether the underlying text token is a part or boundary of the target annotation or not. Figure 1 shows an example tagged sentence. The token *The* is marked as the beginning of a noun phrase ([NP) that ends at *fox* with NP]. The standard approach is to train one separate classifier for each annotation type, i.e. one for noun phrases, one for verb phrases etc. This subproblems can be solved using a multiclass classification algorithm that is trained to predict for each token exactly one class. We can often observe an overlapping of the annotations of the different types in real world applications, such as in chunking, syntactic parsing or ontology based information extraction (OBIE). The token *The* e.g. is at the same time a determiner and the beginning of a noun phrase, which is indeed linguistically a reasonable coincidence. The traditional approach ignores this co-occurrence and is therefore not able to exploit the additional information, namely that determiner are often also noun phrases beginnings.

The approach presented in this work therefore reformulates the many individual multiclass problems into only one multilabel classification problem. In contrast to multiclass, multilabel classification allows an instance to be associated to several classes, in this context often called labels. This means that a token is now allowed to have simultaneously several classes assigned. The purpose of this representation is twofold: on the one hand we obtain a more natural, compact and consistent representation of the extraction problem. On the other hand, the main purpose of this formulation is to allow an underlying multilabel algorithm to exploit relations in the labels such as co-occurrence, exclusions and implications and hence improve the prediction quality. This work evaluates several multilabel learning algorithms and compares them on a dense annotation dataset.

Only few attempts have been made on this subject so far. McDonald *et al.* [2005] were able to improve accuracy in extracting non-contiguous and overlapping segments of different types using an adapted multilabel algorithm. However, their algorithm is not directly comparable since it is centered and adapted to sentences whereas the approach presented here is token based and usable with any multilabel algorithm.

2 Preliminaries

The transformation of an information extraction task into a classification problem was already sketched in Section 1. The next two paragraphs give a more detailed description of the two main processing steps (mainly based on [Loza Mencía, 2009]), continued by the introduction of the employed multilabel classification algorithms (a recent overview can be found in [Tsoumakas *et al.*, 2009]).

Boundary classification In this work we employ the simple but effective *Begin/End* approach. The start and the end of each annotation, i.e. only the boundaries, are marked with the tag *BEGIN* or *END*, the rest is marked as negative examples with *NEG*. The bottom two rows in Figure 1 shows for each type DT (determiner), JJ (adjective), NN (noun) etc. the beginning and ending of an annotation.[1]

In the standard approach, a problem appears when an annotation only includes one token, such as for DT. This would make it necessary to tag a token as *BEGIN* and *END* simultaneously. As this would require a multilabel capable underlying classifier, the common approach is to include an additional class *UNIQUE* which represents both classes at the same time (see also Section 3).

We chose a pragmatic way for solving inconsistencies during the reconstruction of the annotations on the test set: we search for the first appearing of an opening tag and continue the extraction until the first matching closing tag is found, the remaining tags are ignored.

Feature Generation The boundary classification step generates the class information for each training instance, but up to now these instances are empty. The simplest features which can be added are the occurrences of the different tokens themselves. Since the focus of this work lies on the comparison of the classification algorithms, we use only these simple features and refrain from using sophisticated linguistic features. For the same reason we ignore the classification history.

[1] The negative class is omitted since this is the multilabel representation.

token	The	quick	brown	fox	jumps	over	the	lazy	dog
token	the=1	quick=1	brown=1	fox=1	jumps=1	over=1	the=1	lazy=1	dog=1
features	+1.quick=1	+1.brown=1	+1.fox=1	+1.jumps=1	+1.over=1	+1.the=1	+1.lazy=1	+1.dog=1	
		-1.the=1	-1.quick=1	-1.brown=1	-1.fox=1	-1.jumps=1	-1.over=1	-1.the=1	-1.lazy=1
POS	[DT, DT]	[JJ, JJ]	[JJ, JJ]	[NN, NN]	[VBZ, VBZ]	[IN, IN]	[DT, DT]	[JJ, JJ]	[NN, NN]
syntactic	[NP			NP]	[VP	[PP	[NP		NP], PP], VP]

Figure 1: Transformation of a text sentence into a classification problem. Each column shows a token and exemplarily the generated features with a context of one word and the class information of the corresponding generated classification instance. The first row of the class information 'POS' shows the part-of-speech annotations, the second the syntactic annotations given to the token. Abbreviations according to [Marcus *et al.*, 1993]. A '[' denotes the *BEGIN* and ']' the *END* of the corresponding annotation type.

The token features row in Figure 1 shows the type of windowing we used.

2.1 Multilabel Classification

We represent an instance or text position as a vector $\bar{\mathrm{x}} = (x_1, \ldots, x_M)$ in a feature space $\mathcal{X} \subseteq \{0,1\}^N$. In multilabel problems, each instance $\bar{\mathrm{x}}_i$ is assigned to a set of relevant labels y_i, a subset of the K possible classes $\mathcal{Y} = \{c_1, \ldots, c_K\}$, in contrast to multiclass problems, where each instance is mapped to exactly one class, i.e. $\|y_i\| = 1$.

Binary Relevance In the binary relevance (BR) or one-against-all (OAA) method, a multilabel problem with K possible classes is decomposed into K binary problems. For each subproblem, a binary classifier is trained to predict the relevance of the corresponding class.

QWeighted Calibrated Label Ranking QCLR is a recently proposed algorithm, which is an efficient approach for multilabel classification. This algorithms combines three aspects: the pairwise decomposition of multilabel problems, calibrated label ranking for determining multilabel result and an adaption of the QWEIGHTED algorithm for efficient prediction Loza Mencía *et al.* [2009]. In the pairwise decomposition approach, one classifier is trained for each pair of classes, i.e., a problem with K different classes is decomposed into $\frac{K(K-1)}{2}$ smaller subproblems. An example is added to the training set $c_u vs.c_v$ if c_u is a relevant class and c_v is an irrelevant class or vice versa, i.e., $(c_u, c_v) \in y \times \overline{y} \cup \overline{y} \times y$ with $\overline{y} = \mathcal{Y} \backslash y$ as negative label set. During classification, each base classifier is queried and the prediction is interpreted as a vote for one of its two classes. The resulting ranking of classes is split into relevant and irrelevant classes via calibration. The QWEIGHTED approach allows to reduce the classification costs from quadratic to log-linear time.

Label Powerset In the label powerset approach (LP), a meta multiclass problem is constructed where each appearing label combination y_i is interpreted as one separate class. The meta problem is then solved with a normal multiclass algorithm or with the previously presented decomposition methods. In the worse case, the resulting multiclass problem has an increased amount of classes of $min(M, 2^K)$ where M is the number of training examples, however the number tends to be much smaller due to class correlations.

3 Multilabel Classification for Information Extraction

As already outlined, one of the advantages of multilabel classification is the more natural representation since we do not have to work with tricks. Note e.g. that in the *BEGIN/END/UNIQUE* scheme the learning algorithm is forced to learn to distinguish between *UNIQUE* and *BEGIN* resp. *END* though *UNIQUE* is actually a subset of these two classes. This makes this tagging scheme especially interesting for the usage of multilabel classification. Furthermore, the traditional methods do not permit to exploit relations between several annotation types since each type is by design necessarily learned separately. We present therefore in the following the standard approach together with the multilabel alternatives.

Traditional multiclass approach A multiclass classifier is trained for each annotation type, the *BEGIN/END/UNIQUE/NEG* scheme is used. I.e. for each annotation type $a \in A$ a classifier is trained with instances mapped to exactly one class c in $\mathcal{Y}_a = \{\mathit{BEGIN, END, UNIQUE, NEG}\}$. The multiclass problem is solved via one-against-all decomposition in our case.

Binary Relevance and QCLR The extraction task is transformed into one multilabel problem where each token is assigned to a subset y of $\mathcal{Y} = A \times \{\mathit{BEGIN, END}\}$, with A as the set of annotation types. Note that it is not necessary to include the *NEG* as the algorithm is able to predict the empty set.

In the binary relevance setting, the algorithm is not expected to improve from label co-occurrences since each base classifier is trained separately. However, the pairwise approach is at least potentially able to detect non co-occurrences, since we train for each pair of classes a base classifier with instances where the one class is positive and the other negative, i.e. the base classifier is trained with cases where both classes are mutually exclusive. Therefore this approach may be able detect that a co-prediction of two classes is wrong for a determined instance, in contrast to BR, where a base classifier cannot state anything else than relevant or non-relevant. Recently, promising advances were made in enhancing the pairwise approach by the detection and exploitation of present constraints on the labels, such as co-occurrences [Park and Fürnkranz, 2008]. We are currently working on incorporating these ideas.

Label Powerset The multilabel problem is retransformed into a multiclass problem, i.e. the possible classes of a token are in $\mathcal{Y} = 2^{A \times \{BEGIN,END\}}$. Note that for only one annotation type this corresponds to the traditional multiclass approach, since we would obtain $c \in \mathcal{Y} = \{\{\}, \{BEGIN\}, \{END\}, \{BEGIN{,}END\}\}$, which corresponds to $\{BEGIN,\ END,\ UNIQUE,\ NEG\}$. But for more than one annotation type, co-occurrence and implications can effectively be exploited and detected since these co-occurrences are explicitly used as training information. However, the granularity of this information is limited, since the approach is only able to abstract from the co-occurrence of two classes if there is no other class appearing since this would not generate the desired meta co-occurrence class anymore.

4 Evaluation

Since it is difficult to obtain densely annotated (free) corpora e.g. from the field of OBIE, we decided to generate our own dense dataset with the help of the Stanford Parser, which returns the syntactic structure of a sentence.[2] The result of the parser was considered to be the true and correct labeling of the corpus. The first six (scientific) texts from the *Learned* category of the Brown Corpus [Francis and Kucera, 1979] were annotated with this tool, taking the first three documents for training and the remaining for testing. The resulting training set contains 6790 instances and 48 different annotations types, 7091 instances remained for testing. Since each annotation type leads to two tags denoting the *BEGIN* and the *END*, we obtain 96 different labels for the multilabel problem. In average there are 3.34 labels associated per token. For the label powerset representation, 334 classes were retrieved. A window size of 5 resulted in less than 7000 different features. We used the well known LibSVM library with linear kernel and standard settings as our base learner [Chang and Lin, 2001].

The results are shown in Table 4. The first observation is that the multilabel approaches (QCLR and BR) seem to slightly outperform the traditional multiclass approach in terms of F1, and that ML has a slight advantage over LP. A closer look reveals that recall and precision highly depend on the used transformation approaches. LP seems to boost recall while ML and especially the classical multiclass approach improve precision, always at the expense of the opposite measure. The MC setting appears to generate quite conservative classifiers, since the MC extractor predicts 18% less annotations than ML-QCLR and even 28% less than LP-PC.

Note that the absolute values may seem generally low, but remind that these results were produced without linguistic or any other intelligent preprocessing. Moreover, only exact annotation matches were taken into account, counting token matches improves the rates to around 70%.

5 Conclusions

We have presented the approach of presenting an information extraction problem as one multilabel classification problem rather than several independent multiclass problems. This view is more natural to the extraction problem and furthermore potentially allows the exploitation of relations and correlations between overlapping annotations.

Although all multilabel approaches achieve higher F1 scores in the experiments than the standard approach, a direct comparison of both approaches shows up to be difficult, since the traditional approach is focused on precision while the multilabel approaches obtained higher recall to the extend of the precision. Evaluations on more corpora and perhaps with a more extensive also linguistic preprocessing are planned in order to obtain a clearer picture. Nevertheless, it has been demonstrated that the formulation as multilabel problem is at least comparable, particularly considering that the employed algorithms are not especially adapted or designed in order to exploit class correlations. Recent advantages in the relatively new field of multilabel classification let us expect substantial improvements (cf. [Tsoumakas *et al.*, 2009]). Furthermore, we are currently investigating an adaption of the pairwise approach that benefits from restrictions and constraints on the possible class constellations [Park and Fürnkranz, 2008].

Algorihm	Precision	Recall	F1
MC	74.21	34.32	46.93
ML-BR	72.49	40.52	51.98
ML-QCLR	71.49	40.18	51.44
LP-BR	59.67	43.46	50.29
LP-PC	65.12	41.73	50.87

Table 1: Prediction quality of the different algorithms based on exact annotation matches, micro-averaged over all annotation types. MC for the traditional multiclass algorithm, ML for the multilabel transformation and LP for label powerset. PC denotes the pairwise decomposition approach for multiclass problems.

6 References

Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

W. N. Francis and H. Kucera. Brown corpus manual. Technical report, Department of Linguistics, Brown University, Providence, Rhode Island, US, 1979.

Eneldo Loza Mencía, Sang-Hyeun Park, and Johannes Fürnkranz. Efficient voting prediction for pairwise multilabel classification. In *Proceedings of the 11th European Symposium on Artificial Neural Networks (ESANN-09)*, 2009.

Eneldo Loza Mencía. Segmentation of legal documents. In *Proceedings of the 12th International Conference on Artificial Intelligence and Law*, pages 88–97, 2009.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2):313–330, 1993.

Ryan T. McDonald, Koby Crammer, and Fernando Pereira. Flexible text segmentation with structured multilabel classification. In *Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, 2005.

Sang-Hyeun Park and Johannes Fürnkranz. Multi-label classification with label constraints. In *Proceedings of the ECML/PKDD-08 Workshop on Preference Learning (PL-08)*, pages 157–171, 2008.

Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. Mining multi-label data. In *Data Mining and Knowledge Discovery Handbook*. Springer, 2009.

[2] http://www-nlp.stanford.edu/software/lex-parser.shtml