Cost-Sensitive Learning in Answer Extraction

Michael Wiegand[†], Jochen L. Leidner[⋆], and Dietrich Klakow[†]

[†]Spoken Language Systems, Saarland University, Germany
[⋆]Research & Development, Thomson Legal & Regulatory, St. Paul, Minnesota, USA[‡]
Michael.Wiegand@lsv.uni-saarland.de, Jochen.Leidner@Thomson.com, Dietrich.Klakow@lsv.uni-saarland.de

Abstract

One problem of data-driven answer extraction in open-domain factoid question answering is that the class distribution of labeled training data is fairly imbalanced. This imbalance has a deteriorating effect on the performance of resulting classifiers. In this paper, we propose a method to tackle class imbalance by applying some form of *cost-sensitive learning* which is preferable to *sampling*. We present a simple but effective way of estimating the misclassification costs on the basis of the class distribution. This approach offers three benefits. Firstly, it maintains the distribution of the classes of the labeled training data. Secondly, this form of meta-learning can be applied to a wide range of common learning algorithms. Thirdly, this approach can be easily implemented with the help of state-of-the-art machine learning software.

1. Introduction

One problem of data-driven answer extraction in open-domain factoid question answering (QA) is that the class distribution of labeled training data are fairly imbalanced. (Drummond and Holte, 2005) show that, in general, this imbalance has a deteriorating effect on the performance of resulting classifiers. This effect can be very drastic in answer extraction. Our initial answer extraction algorithm using a standard learning algorithm produced only a very small proportion of *true positives* (7 out of 203). Usually, this problem is avoided by applying *sampling*, by which the class distribution usually gets distorted.

In this paper, we propose a more natural way to tackle class imbalance by applying some form of *cost-sensitive learning*. We present a simple but effective way of estimating the misclassification costs on the basis of class distribution. This approach offers three benefits. Firstly, it maintains the distribution of the classes of the labeled training data. Secondly, this form of *meta-learning* can be applied to a wide range of common learning algorithms. Thirdly, this approach can be easily implemented with the help of state-of-the-art machine learning software, such as WEKA (Witten and Frank, 2005).

2. Related Work

Answer extraction is a binary classification problem. Fortunately, most research on learning with imbalanced class distribution deals with this classification type. All standard learning methods suffer from the effects caused by imbalanced class distribution (Drummond and Holte, 2005).

A popular solution to this problem is *sampling*. The two most common types are *down-sampling*, where some of the training instances of the majority class are discarded so that the class distribution is balanced. *Up-sampling*, conversely, establishes this balance by duplicating some training instances of the minority class. There are no definite results as to the supremacy of one type. (Drummond and Holte, 2003) report better results for downsampling for decision trees. (McCarthy et al., 2005) come to the opposite conclusion. The fact that down-sampling actually ignores some labeled data is particularly controversial when it comes to very small training sets. (Chan and Stolfo, 1998) propose partitioning the majority classes to n samples so that each partition is approximately of the size of the minority class. For each of the new resulting n training sets an independent classifier is learned. The final classifier combines the individual classifiers by some sort of meta-learning. Though this method overcomes some of the problems encountered with simple sampling methods, it is fairly processing-intensive.

Cost-sensitive learning (Elkan, 2001) supersedes sampling methods in that it does not alter the original distribution of classes. Since some state-of-the-art toolkits already support this meta-learning, it should be fairly easy to implement a classifier using this approach. According to (McCarthy et al., 2005), the results of cost-sensitive learning are comparable with or even outperform sampling methods. To the best of our knowledge, there has not been any application of cost-sensitive learning to answer extraction.

3. Method

Before describing our proposed method of cost-sensitive learning, we will briefly discuss the answer extraction algorithm we use.

3.1. Answer Extraction in Context

The task of open-domain question answering (QA) as proposed by TREC (Voorhees and Harman, 2005) is to retrieve answers from a text collection given a natural language question, such as *When was Mozart born?* The text collection is usually a large corpus, for example, a collection of newspaper articles. The answers to be retrieved are tiny spans of texts known as *answer snippets* which are extracted from a set of passages deemed to be relevant to a given question. The most prominent questions which are usually asked are *factoid questions*, i.e. questions asking for simple facts. The corresponding semantic types of the

[‡]This research was carried out while the second author was a research associate at Spoken Language Systems, Saarland University.

answer entities, typically dates, locations, person or organization names, are, therefore, well-defined.

A generic QA system is illustrated by Figure 1. First, a question undergoes an analysis in order to determine its question type. By that, we understand the semantic type that the question asks for. Followed by that, the specific question is transformed to a query for a search engine optimized for QA which retrieves documents deemed to be relevant for the question. In order to limit the search, the set of the most relevant passages[1] are retrieved from these documents.

The final step known as *answer extraction* consists of extracting the most likely answer snippet(s) for the given question. Usually, this requires some heavy linguistic analysis by which question and potential answers are compared. In our experiments we consider all noun-phrases with the exception of anaphoric expressions within the set of candidate answer passages as the set of candidate answers or, more precisely, candidate answer constituents.

In a data-driven model, which we discuss in this work, the task of answer extraction can be reformulated as finding an optimal mapping from the question constituent qc (the phrase constituent comprising the interrogative pronoun) to an answer constituent ac_i of a relevant answer passage. Table 1 lists all candidate answer constituents for question-answer Pair (1)-(2)[2]:

(1) Who won the Super Bowl XXXIV?

(2) Tennessee quarterback Steve McNair (9) is brought down by St. Louis' Grant Wistrom (98) in the Rams' 23-16 victory in Super Bowl XXXIV on Sunday.

The classifier to be built for answer extraction regards each possible pair of question constituent, i.e. in our example *Who*, and answer candidate as a training instance x_i. Each instance x_i has a unique class label y_i with $y_i \in \{c_0, c_1\}$ where c_0 is the label for the mappings which are incorrect and c_1 is the label for the mappings which are correct. In our example, there is only one correct mapping (*Who*,*Rams*), but several incorrect mappings, such as (*Who*,*Steve McNair*) or (*Who*,*Super Bowl XXXIV*).

Each training instance is encoded as a set of features which describe the similarity of each tuple (qc, ac_i) on various linguistic and non-linguistic levels. A list of some important orthographic, syntactic and semantic features we use is shown in Table 2. A more detailed discussion of the algorithm and its corresponding features can be found in (Wiegand, 2007).

3.2. Cost-Sensitive Learning Applied to Answer Extraction

The question-answer pair from the previous section illustrated by Table 1 exemplifies the inherent imbalance of the two classes in the entire training set. The class-imbalance we observe is not just an artefact of the particular data-set but are inherent in any open-domain QA dataset. There are always far more incorrect answer constituents than there are correct answer constituents. For training a classifier each of the labeled answer constituents is valuable. Since the negative constituents are a very inhomogeneous set of instances, it is not advisable to ignore any of them for the sake of preserving a balanced class distribution.

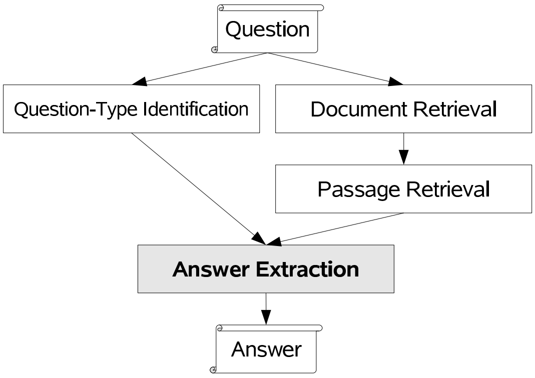

Figure 1: A Generic Architecture of a QA System.

correct answers	Rams
incorrect answers	Tennessee, quarterback, Steve McNair, St. Louis, Grant Wistrom, victory, Super Bowl XXXIV, Sunday

Table 1: Possible Answer Constituents in Answer Passage.

In a formal notation the class imbalance in answer extraction can be expressed as $f(c_0) > f(c_1)$ where $f(c_i)$ is the frequency f of all training instances labeled with class c_i. In the current case, c_0 is the majority class and c_1 the minority class. Resulting classifiers that do not address this imbalance will produce solutions where the proportion of actual negative instances correctly classified is fairly high, i.e. the number of *false positives* (FP) is low. On the other hand, the greater the imbalance of the classes is, the more increases the proportion of actual positive instances incorrectly classified, i.e. the *false negatives* (FN).

In cost-sensitive learning, different costs are assigned to the different types of misclassifications, i.e. FP and FN. Not the solution with the minimal error but the classifier which minimizes the total costs is learned. In our current example, the costs for FN (CFN) should be greater than the costs for FP (CFP). The basic problem is to find a good ratio $CFN : CFP$ given a specific classification problem. Unfortunately, there does not exist a commonly accepted way how to estimate the optimal cost ratio. An appropriate method would need to take both the distribution of the training data and possible biases of the classifier to be used into account.

We propose an ad-hoc method to determine cost ratios on the grounds of class ratios, i.e. CFP is set to $f(c_1)$ and

[1] In our experiments, we used a fixed passage size of one sentence.

[2] As a simplification for illustration we only display one answer passage for the question. In practice, one usually considers a set of answer passages for each question.

Orthographic Features
How similar are the surface strings of the two constituents?
How similar are the surface strings of the two main predicates of the two constituents?
Syntactic Features
Do the constituents have the same grammatical function with regard to their respective main predicate?
How similar is the distance of the two constituents to their respective main predicate?
Do the constituents have the same orientation to their respective main predicate?
Do the heads of the two constituents have the same part-of-speech tag?
Do the main predicates of the two constituents have the same part-of-speech tag?
Semantic Features
How similar are the senses of the heads of the two constituents (use synset relation in WordNet[3])?
How similar are the senses of the main predicates of the two constituents (use synset relation in WordNet)?

Table 2: A Selection of Important Features Used in Answer Extraction . (All features describe the similarity between the question constituent qc and a candidate answer constituent ac_i.)

CFN to $f(c_0)$. We do not claim that this solution is optimal but it is a solution which produces satisfactory results. It may be ad-hoc but its plausibility can be illustrated by a simple example. Imagine a training set with $f(c_1) : f(c_0)$ of $1 : 10$. By setting CFN to 10 and CFP to 1, one actually states that a misclassification of an instance of the majority class c_0 weighs ten times less than an instance of the minority class c_1 because there are ten times more training instances of the majority class. The usage of cost-sensitive learning, thus, weights each training instance by its relative importance.

Considering that standard machine learning toolkits, such as WEKA[4], include cost-sensitive learning as a meta-learner to wrap around standard learning methods, the implementation of this method is very easy and efficient.

4. Evaluation

Our answer extraction classifier is built using the TREC 14 QA Collection (Voorhees, 2005). All results we state below are based on averaged 10-fold cross-validation. The cost-sensitive meta-learner embeds a base learner, we first look at logistic regression here. Two classifiers are built, one comprising only the bare logistic-regression learner and one embedding this learner into a meta-learner being the type of cost-sensitive learning as proposed in the previous section. The corresponding confusion matrices are shown in Table 3. The simple classifier only classifies 7 out of 203 positive instances correctly. For answer extraction, this classifier is useless. The application of cost-sensitive learning sees a significant rise in the number of positive instances to be classified correctly (from 7 to 177). Of course, the improvement on the classification of the minority class goes at the expense of the performance on the classification of the majority class. The number of negative instances classified incorrectly rises from 7 to 2939. This number may appear fairly high, but one should consider that incorrectly classified negative instances weigh far less than incorrectly classified positive instances. Taking the distributional relation of these two classes into account, which has been $1 : 59$ in this training set, one could say that the 2939 misclassified negative instances weigh as much as approximately 50 misclassified positive instances which is a much more reasonable number.

Alternatively, Table 4 displays the difference in performance of logistic regression in terms of precision and recall. While precision drops significantly (by a factor of approximately 8) by our approach, recall improves by factor 25.

But does this result generalize across different types of classifiers? Going beyond linear regression only, Table 4 shows the results of our method on further types of classifiers. We chose a representative of each popular group of classifiers, i.e. one memory-based classifier (Nearest Neighbor), one generative classifier (Naive Bayes), one decision tree (Random Forest), and one discriminative classifier (Logistic Regression). All classifiers display the same behavior, therefore, we conclude that our method is universally applicable to any common base learner.

One could even argue the we should have omitted standard F-Score (F1) which weighs precision and recall equally from the table since this measure heavily favors classifiers with a bias for the majority class. Thus, the improvement by cost-sensitive learning is not necessarily reflected by this measure. According to F1, only Nearest Neighbor and Logistic Regression show an improvement by applying cost-sensitive learning. Too much emphasis is put on precision in this measure. If we, however, look at alternative F-Scores, for instance F2, which weighs recall twice as much as precision, all classifiers show a significant increase in performance. Again, we do not claim that F2 is the optimal evaluation measure for our task. Since the results on F2 are promising, however, we believe that a measure favoring high recall over high precision is appropriate. We leave a formal account of such a measure for future work.

Overall, this experiment shows that our approach produces a far better classifier for our task than the application of a standard learning algorithm.

5. Discussion

We could show that cost-sensitive learning can improve the performance of an answer extraction classifier significantly, however, we have not yet answered what relation our pro-

[4] `http://www.cs.waikato.ac.nz/ml/weka/`

	Precision		Recall		F1		F2	
Measure	**No Cost**	**Cost**	**No Cost**	**Cost**	**No Cost**	**Cost**	**No Cost**	**Cost**
Nearest Neighbor	0.322	0.240	0.276	0.414	0.297	0.304	0.215	0.333
Random Forest	0.400	0.148	0.197	0.616	0.264	0.238	0.237	0.299
Naive Bayes	0.119	0.073	0.172	0.729	0.141	0.133	0.149	0.182
Logistic Regression	0.500	0.065	0.034	0.857	0.065	0.120	0.049	0.169

Table 4: Comparison of Performance of Different Classifiers between Base Learner (*No Cost*) and Cost-Sensitive Learner (*Cost*).

No Cost		**Predicted Class**	
		yes	**no**
Actual Class	**yes**	7	196
	no	7	11929

Cost		**Predicted Class**	
		yes	**no**
Actual Class	**yes**	177	26
	no	2939	8997

Table 3: Comparison of Confusion Matrices of a Logistic Regression Classifier for Answer Extraction between Base Learner (*No Cost*, top) and Embedded Cost-Sensitive Learner (*Cost*, bottom).

posed assignment of cost ratios bears to an optimal assignment. We believe that the optimal assignment can only determined empirically. This would entail checking any possible cost-assignment and considering the assignment which optimizes an appropriate objective function to be optimal. As already indicated in Section 4. we believe that an F-score with an appropriate weighting between precision and recall might serve as an objective function, but we leave the formal account of such a measure for future work. Even if an appropriate objective function can be formally expressed, and the optimal cost-assignment be found[5], the method to find it would lose much of the simplicity of our ad-hoc approach and, therefore, make it less appealing to developers.

6. Conclusion & Future Work

We proposed a novel method to improve classifier learning for answer extraction on data with imbalanced class distribution by embedding learning methods into a cost-sensitive meta-learner. Our evaluation shows that such a classifier containing a cost ratio for the different classification errors derived from the class distribution clearly outperforms a standard learning algorithm.

Acknowledgments

Michael Wiegand was funded by the German research council DFG through the International Research Training Group "IRTG" between Saarland University and University of Edinburgh. This work was funded in part by BMBF under contract 01 IM D01 M.

7. References

Philip Chan and Salvatore Stolfo. 1998. Toward Scalable Learning with Non-uniform Class and Cost Distributions: A Case Study in Credit Card Fraud Detection. In *Knowledge Discovery and Data Mining*, pages 164–168, Menlo Park, CA, USA. AAAI.

Chris Drummond and Robert Holte. 2003. C4.5, Class Imbalance, and Cost Sensitivity: Why Under-Sampling Beats Over-Sampling. In *Workshop on Learning from Imbalanced Datasets*, Washington, DC, USA.

Chris Drummond and Robert Holte. 2005. Severe Class Imbalance: Why Better Algorithms Aren't the Answer. In *Proceedings of the 16th European Conference of Machine Learning*, Porto, Portugal. NRC.

Charles Elkan. 2001. The Foundations of Cost-Sensitive Learning. In *17th International Joint Conference on Artificial Intelligence*, pages 973–978, Seattle, WA, USA.

Kate McCarthy, Bibi Zabar, and Gary Weiss. 2005. Does Cost-Sensitive Learning Beat Sampling for Classifying Rare Classes? In *Proceedings of the 1st International Workshop on Utility-based Data Mining*, pages 69–77, Bronx, NY, USA. ACM Press.

Ellen Voorhees and Donna Harman. 2005. *TREC: Experiment and Evaluation in Information Retrieval*. The MIT Press, Cambridge, Massachusetts, USA.

Ellen Voorhees. 2005. Overview of the TREC-14 Question-Answering Track. In *Proceedings of the 14th Text Retrieval Conference (TREC 2005)*, Gaithersburg, MD.

Michael Wiegand. 2007. Event-Based Modelling in Quesion Answering. Master's thesis, Saarland University, Saarbrücken, Germany, March. `http://www.coli.uni-saarland.de/~miwieg/pub/dipl-thesis.pdf`.

Ian Witten and Eibe Frank. 2005. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Francisco, California, USA, second edition.

[5] We believe that in order to ease the computational time of this approach, the approximation by applying greedy learning in order to speed up the search might be advisable.