Towards Effective Short Text Deep Classification

Xinruo Sun
Apex Data & Knowledge
Management Lab
Shanghai Jiao Tong University
Shanghai, China
xrsun@apex.sjtu.edu.cn

Haofen Wang
Apex Data & Knowledge
Management Lab
Shanghai Jiao Tong University
Shanghai, China
whfcarter@apex.sjtu.edu.cn

Yong Yu
Apex Data & Knowledge
Management Lab
Shanghai Jiao Tong University
Shanghai, China
yyu@apex.sjtu.edu.cn

ABSTRACT

Recently, more and more short texts (e.g., ads, tweets) appear on the Web. Classifying short texts into a large taxonomy like ODP or Wikipedia category system has become an important mining task to improve the performance of many applications such as contextual advertising and topic detection for micro-blogging. In this paper, we propose a novel multi-stage classification approach to solve the problem. First, explicit semantic analysis is used to add more features for both short texts and categories. Second, we leverage information retrieval technologies to fetch the most relevant categories for an input short text from thousands of candidates. Finally, a SVM classifier is applied on only a few selected categories to return the final answer. Our experimental results show that the proposed method achieved significant improvements on classification accuracy compared with several existing state of art approaches.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Retrieval Models

General Terms

Algorithm

Keywords

short text, classification, large scale hierarchy

1. INTRODUCTION

In recent years, with the emergence of the Social Web, more and more short texts such as ads, tags and tweets are generated and consumed by web applications and users. Classifying short texts into a large taxonomy like ODP or Wikipedia category system has become an important mining task to improve the performance of many applications such as contextual advertising and topic detection for micro-blogging. However, this task faces two main challenges. Firstly, the information is expressed in very short text, which results in feature sparseness. Secondly, categories at deep levels of a large taxonomy usually have too few documents to train a classifier with satisfactory performance using statistical learning.

There are two categories of research work related to our task: short text classification and deep classification. The former tries to deal with the feature sparseness issue by leveraging feature expansion while it only proves its effectivess when on a limited number of categories. A repesentiative example[3] classifies tweets into News, Events, Opinions, Deals and Private Messages. The latter category aims at preserving acceptable accuracy when facing a large-scale taxonomy. As mentioned in a report of a recent competition on deep classification[2], all participants focused on large hierarchies instead of situations when the length of textual content is short or categories contain few documents inside. To the best of our knowledge, there is no previous work to solve the above two challenges in the context of short text deep classification.

In this paper, we propose a multi-stage approach for effective short text deep classification. The contributions are threefolds. First we leverage explicit semantic analysis to extract rich explicit concept information from Wikipedia for short text expansion to solve the sparsity issue. Second, documents from descendants of a category in the taxonomy are aggregated to further enrich features. Third, information retrieval techniques are employed to find a few most relevant categories to the input short text for further finer-grained classification, thus avoiding the poor performance of traditional classifiers over a large-scale taxonomy.

2. APPROACH DETAILS

Text documents are usually represented as feature vectors using a mapping function $\Phi : D \rightarrow \mathcal{R}^m$, where D represents a document collection, $\mathcal{R}$ denotes the real-value set, and m is the feature dimension. One common feature representation is tfidf weighting: $\Phi_{tfidf}(d) = (tfidf_d(t_1), \ldots, tfidf_d(t_m))$.

When the lengths of text snippets are very short with several words, the tfidf scheme suffers from feature sparseness. We employ *explicit semantic analysis* (ESA)[1] to tackle this issue. ESA maps natural language fragments into a set of relevant Wikipedia concepts. More precisely, denote the Wikipedia article collection as $W = \{a_1, \ldots, a_n\}$. ESA uses $\Phi_{esa}(d) = (as(d, a_1), \ldots, as(d, a_n))$, where the function as calculates the *association strength* of documents with concepts. A widely used choice is to select tfidf weighting as well: $as(d, a_i) = \sum_{w_j \in d} tf_d(w_j) \cdot tfidf_{a_i}(w_j)$.

We treat categories as virtual documents, thus using the same feature representation scheme as individual documents, by computing the centroids of these categories. Current deep classification methods compute *flat centroids* without considering documents belonging to its descendant categories.

SIGIR'11, July 24–28, 2011, Beijing, China.
ACM 978-1-4503-0757-4/11/07.

Instead, we compute *descendant centroids* that aggregate data from descendants in the hierarchy.

Facing a large number of categories, the accuracy cannot be ensured by directly applying a state of the art classifier. Thus we employ information retrieval techniques to first search for the most relevant categories for an input short text. We apply the cosine similarity to calculate the k nearest categories.

As the result of the search stage, the number of candidate categories significantly reduced to a small amount. So finer-grained classification methods could be employed to achieve better performance. We choose SVM in this paper. For each candidate category, the documents belonging to it become training data. A SVM is trained on these categories. The classifier is then used to classify the test document.

The two kinds of features, tfidf and esa, sometimes result in different classification results. We use an *ensemble classifier* to make the best of both of them. Suppose the classifiers estimate the probability of test sample d being in category c are P_{tfidf} and P_{esa} respectively, a weighted sum is computed as a simple way to ensemble:

$$P_{ensemble} = P_{tfidf} + \alpha P_{esa}$$

In the formula, α is a parameter, which can be determined according to the actual dataset using linear regression.

3. EXPERIMENTS

We used Open Directory Project (ODP)[1] in the experiments. It is a large directory of web pages labeled using a hierarchical category system. We randomly sampled 11,000 pages to be our test collection, the remaining being the training collection containing 1,299,987 pages distributed among 137,489 categories. In the test collection, the titles and descriptions are concatenated to be the test documents. In order to evaluate how shortness of training data can affect different classifiers, we made two training datasets of different text lengths, LITTLE and SHORT. The content of the former is just the title and the latter is the concatenation of the title and the description.

dataset	description	average document length
LITTLE	title of the web pages	3.11
SHORT	title and short description	14.9

Figure 1: Description of training datasets

To construct the ESA index, we obtained the Wikipedia dump of Nov.15, 2010. We selected the Wikipedia articles representing concrete concepts using heuristics similar to [1], resulting in a collection of 1,333,355 concepts.

As mentioned in [2], direct deployment of big-bang approaches and top-down approaches suffers from high computational complexity and propagation errors. Instead, we choosed a two-stage approach as the baseline. First search for k nearest documents using tfidf weighting scheme. Then the same SVM learn-and-classify strategy is carried out.

As a preliminary experiment, we fixed $k = 15$ and $\alpha = 1$. We evaluated the classification performance using Mi-F1. First, we compare the performance between our approach and the baseline on both datasets. One can see in Figure 2 that the ensemble classifier achieves higher Mi-F1 scores than the baseline on almost all levels. Also note that the improvement is not as big on SHORT as on LITTLE. This is because SHORT can not benefit much by enriching features as it already contains quite rich features.

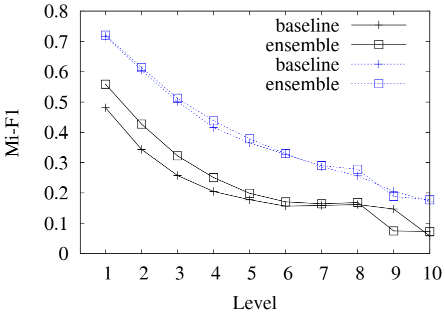

Figure 2: Ensemble classifier compared with baseline. Solid lines are of LITTLE, dotted lines are of SHORT.

Next, we compare the degree to which different features contribute to the performance. The result in Figure 3 shows that the ensemble classifier performs better than both single feature-based classifiers. The esa feature outperforms the ensemble classifier in the higher levels because we used the same α for all levels. This suggests an area of improvement by learning different αs for different levels.

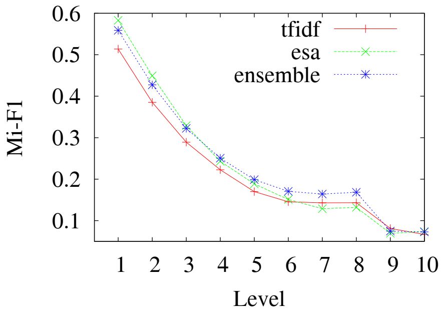

Figure 3: Both features compared with ensemble classifier on LITTLE

4. CONCLUSION AND FUTURE WORK

We conclude that our approach successfully improves the deep classification performance on short text. As of future work, we will try a more advanced ensemble method.

5. REFERENCES

[1] E. Gabrilovich and S. Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *AAAI*, pages 6–12, 2007.

[2] A. Kosmopoulos, E. Gaussier, G. Paliouras, and S. Aseervatham. The ecir 2010 large scale hierarchical classification workshop. *SIGIR Forum*, 44:23–32, 2010.

[3] B. Sriram, D. Fuhry, E. Demir, H. Ferhatosmanoglu, and M. Demirbas. Short text classification in twitter to improve information filtering. In *SIGIR*, pages 841–842, 2010.

[1] http://www.dmoz.org