Large-Scale Hierarchical Text Classification without Labelled Data

Viet Ha-Thuc
Computer Science Department
The University of Iowa
Iowa City, IA, USA
hviet@cs.uiowa.edu

Jean-Michel Renders
Xerox Research Centre Europe
Meylan, France
jean-michel.renders@xrce.xerox.com

ABSTRACT

The traditional machine learning approaches for text classification often require labelled data for learning classifiers. However, when applied to large-scale classification involving thousands of categories, creating such labelled data is extremely expensive since typically the data is manually labelled by humans. Motivated by this, we propose a novel approach for large-scale hierarchical text classification which does not require any labelled data. We explore a perspective where the meaning of a category is not defined by human-labelled documents, but by its description and more importantly its relationships with other categories (e.g. its ascendants and descendants). Specifically, we take advantage of the ontological knowledge in all phases of the whole process, namely when retrieving pseudo-labelled documents, when iteratively training the category models and when categorizing test documents. Our experiments based on a taxonomy containing 1131 categories and widely adopted in the news industry as a standard for the NewsML framework demonstrate the effectiveness of our approach in these phases both qualitatively and quantitatively. In particular, we emphasize that just by taking the simple ontological knowledge defined in the category hierarchy, we could automatically build a large-scale hierarchical classifier with reasonable performance of 67% in terms of the hierarchy-based F-1 measure.

Categories and Subject Descriptors

H.1 [**Models and Principles**]: General; H.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Algorithms

Keywords

Topic Models, Hierarchical Text Classification, Weakly Supervised Classification, Classification with No Labelled Data

WSDM'11, February 9–12, 2011, Hong Kong, China.

1. INTRODUCTION

With the exponential growth of text data, particularly on the Web, hierarchical organization of text documents is becoming increasingly important to manage the data. Along with the widespread use of the hierarchical data management, comes the need for automatic classification of documents to the categories in the hierarchy. Traditional supervised and semi-supervised approaches for hierarchical text classification often require labelled data for learning classifiers. However, when applied to large-scale classification which involves thousands of categories (topics), creating such labelled data, even just a few documents per category, is extremely expensive since typically the data is manually labelled by humans. Motivated by this, we propose a novel approach for large-scale hierarchical text classification which does not require any labelled data.

In this paper, we explore another perspective where the meaning of a category is not defined by human-labelled documents, but by its descriptions and more importantly its relationships with other categories (e.g. its ascendants and descendants). Specifically, we take advantage of the ontological knowledge in all three phases of the whole process. First, we exploit the hierarchy to construct a context-aware query for each category. The query is then submitted to a web search engine to get pseudo-relevant documents for that category. Second, given pseudo-relevant documents for categories, we propose a hierarchical topic model approach to extract a language model (multinomial distribution over words) for each category. Note that in the previous phase, even though we use context-aware queries, the retrieved documents still contain a lot of noise. In the second phase, our hierarchical topic model takes the relationships amongst categories defined in the hierarchy to exclude noise, to identify really relevant parts in training documents, and to estimate category language models from these relevant parts only. Finally, given extracted category language models, the hierarchical structure is again exploited to classify test documents into categories. We propose a novel classification algorithm using information propagated both top-down and bottom-up when making decisions.

We demonstrate the effectiveness of our approach through a series of experiments based on a recent taxonomy released by the IPTC (International Press and Telecommunications Council; see details on *www.iptc.org*), that is more and more used by main news agencies all over the world as a standard for annotating news items and events. This taxonomy includes 1131 categories, organised in a hierarchical tree that contains up to 6 levels including the common root. We show

the benefits of using the ontological knowledge at different stages both qualitatively and quantitatively. In particular, we emphasize that just by taking the simple ontological knowledge defined in the category hierarchy and not using any labelled data, we could automatically build a large-scale hierarchical classifier with reasonable performance. Specifically, we get performance of 67% in terms of the hierarchical version of the F-1 measure, when classifying news items from popular sites (recall that in large-scale classification, particularly in our experiments, the system has to make decisions amongst more than one thousand possible choices).

2. OVERALL FRAMEWORK

In this section, we introduce an overview of our proposed approach. The overall framework is described in Figure 1. First, we exploit the hierarchy to construct an enriched and context-aware query for each category. Basically, for each category, we use its ancestors to define a context for the category and (partially) resolve possible ambiguities. We also exploit its children as special cases to enrich the query. The query is then submitted to a web search engine to get pseudo-relevant documents for the category. We present this phase in more detail in Section 3.

Second, given pseudo-relevant documents for categories, we extract a language model (multinomial distribution over words) for each category. Note that in the previous phase, even though we use enriched and context-aware queries, the retrieved documents are still very likely to contain noise. Therefore, the challenge in the second phase is to exclude noise (non-relevant parts) and identify really relevant parts in training documents. Then, the category language models are estimated from the relevant parts only. To achieve this, we propose a hierarchical topic model extracting a language model for each category by using not only its training documents but also its position in the hierarchy and relationships with other categories. The details of this phase are described in Section 4.

Finally, given extracted category language models, we classify test documents into categories. We propose a novel top-down classification approach taking advantage of the hierarchical structure. To alleviate the risk of cascading error, which is common in previous top-down approaches, our approach softens its decisions at upper levels. Moreover, when making decisions at these levels, the approach also takes into account information propagating from lower levels (bottom-up). The approach is based on a hierarchical extension of the inference algorithm for `LDA` (Latent Dirichlet Allocation), that integrates by construction the document context into word features to resolve the polysemy issue (e.g. word feature *race* is important with different senses with respect to category "*motorcycling*" and "*people*"). Finally, by taking the hierarchical structure into account, the algorithm could prune a large part of the hierarchy from consideration. Therefore, the algorithm scales well when the number of categories increases. The details of this phase are described in Section 5.

3. RETRIEVING TRAINING DOCUMENTS

Basically, for each category, we construct a query that we then submit to a search engine associated to an external resource (typically the Web). We take the top k retrieved documents and temporarily consider these documents as positive examples of the category.

When constructing the queries, we exploit the hierarchical relationships between the categories. In our case, we rely on a recent taxonomy designed to classify news items and events in the journalism and telecommunication industry: the IPTC taxonomy (see *www.iptc.org*). This taxonomy, that is now adopted by an increasing number of news agencies, consists of 1131 categories organised in 6 levels including the root. These categories cover all domains, from arts and culture, to weather forecasts, including crimes, disasters, politics, education, economics, ... For each category, for instance "*security*" [1] (*economic sector/computing and information technology/security*), the upper level categories (e.g. "*computing and information technology*") specify the global context of the category; as such, they are useful to disambiguate with respect to categories having similar (or even the same) titles, for example "*securities*" [2] (*market and exchange/securities*).

On the other hand, we find that given a category, for instance "*economic indicator*" [3], its sub-categories (i.e. the children in the hierarchy) such as "*gross domestic product*", "*industrial production*" are also useful. The sub-categories are special cases of the parent category. So, they could be used to enrich the corresponding query. Given the two observations above, we construct query for each category by combining the title and description of itself with the titles of its parent and children. These queries are then sent to a web search engine. We conduct two searches for each query. For the first one, we search on the general Web, and take the top-50 retrieved documents. For the second one, we limit the search to the Wikipedia site, and take the top-10 retrieved documents. The two results are merged and used as training documents for the category. Our goal in this section is not to rigorously explore the best way to exploit the structure of a specific hierarchy in the query formation or to find out the best relative weights to combine data retrieved from different sources. Instead, we use a rather straightforward approach to implement the intuitions described above, with no particular ad-hoc tuning: we simply give two times more importance to current category description with respect to the parent category description and to the children descriptions.

4. EXTRACTING CATEGORY LANGUAGE MODELS

Given training sets for categories in the hierarchy (pseudo-relevant documents obtained in the previous phase), we will estimate a language model $p(word|category)$ for each of these categories. The challenge in estimating the language models from training documents is that these training documents could also contain portions that are non-relevant to the category. For example, a training document about a "show of a rock band in London" for category "*rock and roll music*" could also contain terms relevant to more general categories such as "*music*" and "*art and entertainment*". It could also contain terms specific to the local context of the document such as *London* or proper names of the bar as well as the

[1]This corresponds to the code=20000229 of the IPTC taxonomy.
[2]IPTC code=20000394
[3]IPTC code=20000358

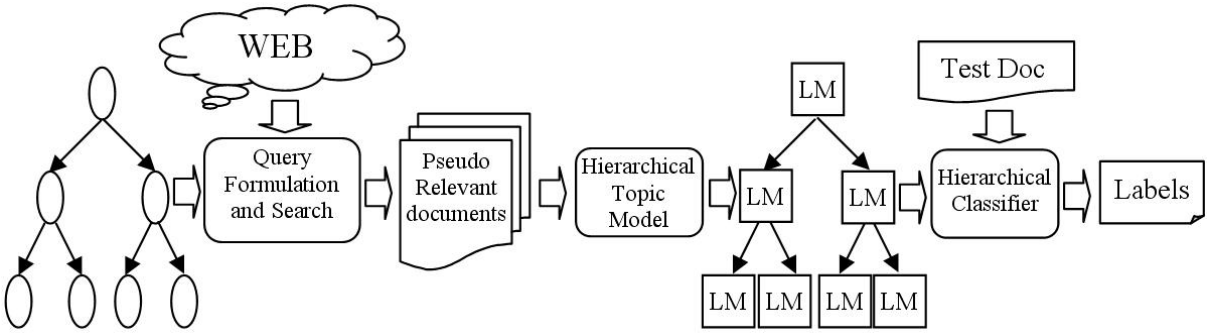

Figure 1: A Framework for Large-scale Text Classification without Labelled Data

band members. Not removing the general terms could make the language model for "*rock music*" highly overlapping and confusing with the language models for its sibling categories such as "*folk music*" or "*country music*". On the other hand, not excluding all document-specific terms could make the language model for the category over-fit to its training set.

It is worth noting that enriching the search query for each category by taking information of its parent and children into account as described in previous section is necessary to reduce ambiguities. On the other hand, this enrichment, however, makes the queries and consequently the training sets of linked categories highly overlapping. So, in the phase of extracting language models from these documents, it is crucial to exclude general terms, especially for low-level categories.

We address this challenge by proposing a hierarchical topic model with ontological guidance for extracting these language models. The approach takes into account the fact that although a document d may be relevant to a category (or, equivalently in this context, a topic[4]) c in the hierarchy, it could still have non-relevant portions. Specifically, a training document d is hypothesized to be generated by a mixture of multiple topics: the category c itself, its ascendant categories explaining general terms (including a "background" topic at root explaining the general English vocabulary), and a document-specific topic $t_o(d)$ responsible for generating terms on other themes also mentioned in the document. These terms are specific to the document context and not relevant to c or its ascendant categories. The contributions of these topics in training documents are automatically inferred and only truly relevant portions (the ones generated by c itself) will contribute to the estimated language model for c. The model has also been shown efficient for modeling news events in the social Web in our recent work [11]. The model description and the inference algorithm are described in detail in the next subsections.

4.1 Hierarchical Topic Model with Ontological Guidance

Hierarchical Topic Model with Ontological Guidance is a generative model describing the process of generating relevant documents for topics in a given hierarchy. Let us denote by W, the number of words in the vocabulary, and by L_c, the level of topic c in the hierarchy ($L_b = 0$ for the background (root) topic). The multinomial distributions (i.e. the language models of the different topics, including the background) are denoted by Φ followed by a subscript that refers to the topic. These multinomial distributions are sampled from a W-dimensional Dirichlet distribution with hyper-parameters β, denoted by $W\text{-}Dir(\beta)$. As any pseudo-relevant document d (for category c) will be modelled as a mixture of multinomial distributions for topics in the path from the root to c itself and a document-specific topic $t_o(d)$, we denote the corresponding mixture weights by Θ_d. Θ_d is sampled from a Dirichlet distribution with hyper-parameters α. The generative process is formally described as follows:

1. Pick a multinomial distribution Φ_b for the background language model from $W\text{-}Dir(\beta)$
2. For each topic c in the hierarchy:
2.1 Pick a multinomial distribution Φ_c from $W\text{-}Dir(\beta)$
2.2 For each document d (pseudo-)relevant to c:
2.2.1 Pick a multinomial $\Phi_{t_o(d)}$ from $W\text{-}Dir(\beta)$
2.2.2 Pick a mixing proportion vector Θ_d for (L_c+2) topics $T_d = \{background \ldots c, t_0(d)\}$ from $(L_c + 2)\text{-}Dir(\alpha)$
2.2.3 For each token in d
2.2.3.1 Pick a topic z in set T_d from Θ_d
2.2.3.2 Pick a word w from Φ_z

The graphical model describing the generative process of training documents belonging a category c is shown in Figure 2. The number at the low-right corner of each box (plate) indicates the number of iterations of that box. $|D_c|$ is the number of training documents related to the current category c, while N_d is the total number of tokens in the current document d. For each document d, word tokens in the document w and $T_d = \{background \ldots c, t_0(d)\}$ containing indices of topics generating d are observed and denoted by shaded circles. Unshaded circles represent latent variables.

Observe that the scope of background topic (root) is common to all training documents. The scope of a topic c in the hierarchy covers documents in the corresponding sub-tree (i.e. training documents associated to the category itself and its descendants, if any). The scope of $t_o(d)$ includes only document d. Therefore, the background category will explain words commonly appearing in all training documents of all categories (e.g. stop words). Each topic c generates words relevant to the top level of the sub-tree it represents (too general words are explained by its ascendants, too specific words are explained by its descendants or $t_o(d)$ topics). In each document d, $t_o(d)$ generates words specific to the context of the document but not relevant to any category

[4]In this work, we suppose that each category corresponds to one topic. Besides categories, topics also include document-specific components and the background language.

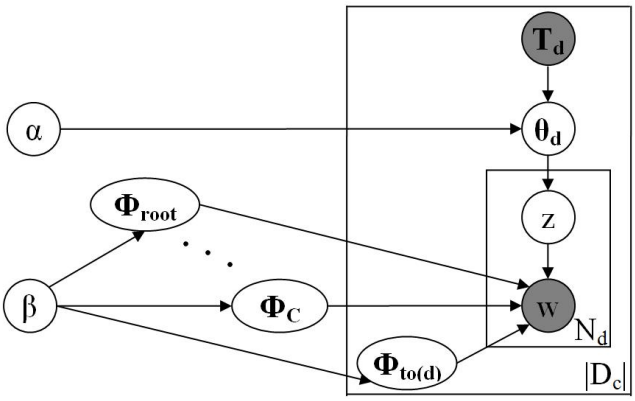

Figure 2: Graphical Model of the Hierarchic Topic Model

1. Initialize variables: $\Phi_z^{(0)}$ and $\Theta_d^{(0)}$ for all topics and all training documents
2. For s = 0 to the desired number of iterations:
   2.1 For each token in a training document d of a category c: Sample its latent topic $z^{(s+1)}$ in the set T={*background* ... c, $t_o(d)$} from distribution:

$$p(z \mid w,d) \propto p(w \mid z)p(z \mid d) = \Phi_{z,w}^{(s)}\ \Theta_{d,z}^{(s)}$$

   2.2 Update $\Phi_z^{(s+1)}$ for all topics:

$$\Phi_{z,w}^{(s+1)} = p(w \mid z) = \frac{m_{z,w}^{(s+1)} + \beta}{\sum_{w'=1}^{W}(m_{z,w'}^{(s+1)} + \beta)}$$

   2.3 Update mixture weights for each training document d:

$$\Theta_{d,z}^{(s+1)} = p(z \mid d) = \frac{n_{d,z}^{(s+1)} + \alpha}{\sum_{z' \in T}(n_{d,z'}^{(s+1)} + \alpha)}$$

$\forall z \in T$, where T={*background* ... c, $t_o(d)$}

Figure 3: Inference Algorithm

from the root to category c to which the document belongs. So, semantic meaning of a category is not only determined by its training documents but also by its relationships to other categories in the tree. All multinomial distributions for categories and category mixing proportions in documents are automatically inferred by the following algorithm.

4.2 Inference Algorithm

Similar to previous works [10, 2], we also apply a variant of Gibbs sampling technique to infer all latent variables (multinomial distributions and mixing proportions in documents) given the observed tokens (a token is a particular occurrence of a word in a document). The algorithm is formally presented in the Figure 3. In Step (1), the parameters of multinomial distributions (Φ_c) are initialized by their maximum likelihood estimates from training documents belonging to the corresponding sub-tree, and each $\Phi_{t_o(d)}$ is initialized by its maximum likelihood estimate from document d. Mixing proportions in all documents are initialized uniformly. In each iteration of Step (2), we sample latent topic generating

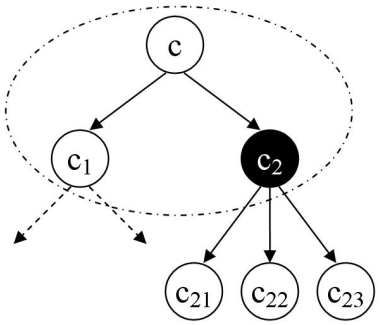

Figure 4: Sampling Example

each token from its posterior (Step(2.1)). After sampling for all tokens, we update the multinomial distributions and mixing proportions (Steps (2.2) and (2.3)), where $m_{z,w}$ is the number of times word w is assigned to topic z, and $n_{d,z}$ is the number of times topic z is assigned to a token in document d. These sampling and updating steps are repeated until convergence. In practice, we set a value for the maximum number of iterations.

5. HIERARCHICAL CLASSIFICATION

In this study, we consider a general case where a test document could be assigned to multiple categories at different levels of abstractions in the hierarchy. This setting is more complicated than the case where each test document is assigned to only the leaf categories, but the setting is more natural in practice. We assume each test document is generated by a mixture of all nodes in the hierarchy (if some category is totally irrelevant to the document, its mixture weight will be close to zero). So, the multi-labeled classification problem can be seen as the task of inferring mixture weights given the document and the language models of all nodes estimated in the previous phase. We solve this inference problem by a sampling approach, keeping the language models fixed. Specifically, we iteratively sample the latent topics generating the tokens in the test document. Then, we rank categories by their mixture weights $p(c|d)$, estimated from the samples.

We exploit the hierarchical structure to decompose the sampling step for each token into sub-steps. The sampling algorithm starts from the root, $c = root$. Assume c has two children c_1 and c_2 (see Figure 4). The algorithm probabilistically decides if the token is generated by c or a node in one of the two sub-trees by sampling in the set $S = \{c, c_1^{subtree}, c_2^{subtree}\}$ (where $c_i^{subtree}$ is a pseudo-topic representing the whole sub-tree rooted at c_i) from posterior distribution as in Equation 1. In this equation, $p(z|d)$ indicates how much z contributes to the content of document d. These probabilities are estimated iteratively (as we will show later). $p(w|c)$ is estimated in the previous phase (Section 4). $p(w|c_i^{subtree})$ is a multinomial distribution representing the language model of the whole sub-tree rooted at c_i . It is estimated from the multinomial distributions of all nodes belonging to the sub-trees including c_i itself (see Equations 2 and 3). When the algorithm samples the latent topic in the set S, if topic c is picked, then the latent topic for the token is determined: it is c. If one of the sub-trees, for instance $c_2^{subtree}$, is picked, i.e. the token is generated by a node in

Input: (i) a topic hierarchy and language models of all topics and of all pseudo-topics corresponding subtrees in the hierarchy
(ii) an unseen document d

Output: p(z| d) for all topics

Algorithm:

1. Initialize p(z| d) uniformly for all topics in the hierarchy
2. Loop for a desired number (N) of iterations
   2.1 For each token in document d:
      2.1.1 Loop for M times:
         2.1.1.1 Current_node = root
         2.1.1.2 Let c_1, c_2 ... c_P be the children of *Current_node* and $c_1^{subtree}$, $c_2^{subtree}$ ... $c_P^{subtree}$ be pseudo-topics representing the corresponding subtrees.
         2.1.1.3 Sample a latent topic for the token in the set $S=\{Current\_node, c_1^{subtree}, c_2^{subtree} \ldots c_P^{subtree}\}$ from the distribution:

$$p(z \mid w,d) \propto p(w \mid z)p(z \mid d), \quad z \in S$$

where $p(c_i^{subtree} \mid d) = \sum_{z \in c_i^{subtree}} p(z \mid d)$

         2.1.1.4 If one of the pseudo topics, for instance $c_1^{subtree}$, is picked, then:
            $Current\_node = c_i$
            Go to Step 2.1.1.2
   2.2 Update p(z| d) for all topics z from samples generated above by maximum likelihood.

Figure 5: Hierarchical Classification Algorithm

the sub-tree rooted at c_2, then the sampling is incomplete and the algorithm has to proceed to this sub-tree (Figure 4) and repeats the process until the latent topic is determined.

$$p(z|w,d) \propto p(w|z)p(z|d), z \in \{c, c_1^{subtree}, c_2^{subtree}\} \quad (1)$$

$$p(w|c_i^{subtree}) = \sum_{z \in c_i^{subtree}} p(w|z)p(z|c_i^{subtree}) \quad (2)$$

$$\approx \frac{\sum_{z \in c_i^{subtree}} p(w|z)}{|c_i^{subtree}|} \quad (3)$$

The classification algorithm is formally described in Figure 5. The mixture weights are initialized uniformly (Step 1) and will be updated iteratively. In Step 2.1, the algorithm samples latent topics of all tokens in the test document from the corresponding posterior distributions. To avoid the issue of cascading errors, the algorithm "softens" its behaviour by doing the sampling M times (Step 2.1.1). As described earlier, this sampling step is performed in a top-down manner starting from the root (Step 2.1.1.1). When sampling (Step 2.1.2.3), the topic mixing proportions $p(z|d)$ are integrated in the posterior probabilities. This factor representing the context of document d aims to resolve word ambiguity. For example, if d is an article about a fishing resort, then terms in d like "*fish*", "*fishing*" or "*boat*" have high likelihood $p(word|topic)$ in both topics "*travel*" and "*fishing industry*". However, by taking the context of the document into account, the algorithm can recognize that these terms are not meant to be mentioned in the context of topic "*fishing industry*". After generating M samples for all tokens, the algorithm re-updates the mixture weights (Step 2.2). The whole process (including Steps (2.1) and (2.2)) is iterated N times. M and N are parameters.

In the hierarchical sampling process above (from Steps 2.1.1.1 to 2.1.1.4), a token is assigned to topic c only if it is also assigned to all sub-trees rooted at ancestors of c. On the other hand, when the algorithm decides to assign a token to a sub-tree, the algorithm takes information from all the nodes in the sub-tree into account (recall how $p(w|c_i^{subtree})$ is estimated in Equation 3). So, when sampling at a particular level in the hierarchy, the algorithm uses information propagated both top-down and bottom-up to alleviate possibly inaccurate estimations of probabilities $p(w|c)$ for some words w and categories c. Moreover, by hierarchically decomposing the sampling, the algorithm can prune a large part of the hierarchy from consideration in the sampling process. As a result, the number of nodes it considers is only $O(log(n))$, where n is the number of categories in the hierarchy. Therefore, it scales well when then number of categories increases (as in the case of large-scale classification).

It is worth noting that the way we exploit top-down and bottom-up information is different from previous works. McCallum et al.[18] and Toutanova et al. [21] use top-down information, and Wetzker et al. [24] use bottom-up information to smooth estimations $p(w|c)$ for all categories. Though the smoothing could make the distributions less sensitive to noise, it has a side effect that makes distributions of similar topics which share common ancestors or descendants highly overlapping and less distinguishable. In our approach, when the algorithm already reaches the node c, as in the example above, and makes a choice amongst c, $c_1^{subtree}$ or $c_2^{subtree}$, the information of ancestors of c is no longer needed since it is the same for all of the three. The multinomial distributions of the categories should not waste their probability mass on the common features, and instead should focus on features distinguishing each of the categories with the rest. If $c_2^{subtree}$ is sampled, the algorithm proceeds in this direction. At that time, it then uses the multinomial of c_2 itself, not $c_2^{subtree}$ (i.e. the bottom-up information is taken out) to distinguish category c_2 from its children ($c_{2.1}$ and $c_{2.2}$). So, the algorithm uses both top-down and bottom-up information in an adaptive way to alleviate the problem of noise in topic language model estimations as well as to maintain discriminative power of these language models.

6. EXPERIMENTS

In this section, we demonstrate the effectiveness of our approach in estimating category language models and in classifying test documents. We first describe the topic hierarchy and test documents we use in our experiments. Then, we present performances of our approach in each of the two phases in comparison with baselines.

6.1 Topic Hierarchy and Test Set

As already mentioned, the IPTC (*International Press and Telecommunications Council*) has recently released a taxonomy of codes, for annotating news items and events. It is becoming a standard for main news agencies and an important component of the `NewsML` standard as media-independent structural framework for multi-media news. This taxonomy contains 1131 categories, organised in a tree that contains

up to 6 levels including the common background (root). The first level contains 17 main topics, covering domains such as business, economics, education, religion, crimes, disasters, weather, etc. The last level contains very specific topics, such as "assisted suicide" or "methodist christians". The average number of children is around 3 in this hierarchy. Each category contains a title (typically two or three words), as well as a short description (25 words on average).

The evaluation set consists of a collection of 1130 news items[5], crawled on the web sites of 4 news agencies (CNN, Reuters, France24 and DW-World), during the first two weeks of June 2010. The preprocessing consisted in cleaning the html files (boilerplate removal, etc.), and removing stopwords. Two independent annotators (with a journalism background) labelled this set of 1130 news items: for each item, they were allowed to give as many labels as they wanted, provided that they used the most specific ones in the trees.

6.2 Extracting Category Language Models

In this subsection, we show the effectiveness of our approach in estimating category language models by comparing with the standard maximum likelihood approach (where the language model of a category is derived from the count of the total number of occurrences of a particular word divided by the total number of tokens, when we consider the concatenation of all documents related to the category). The two approaches take the IPTC topic hierarchy and pseudo-relevant documents as described in Section 3 as inputs. Figures 6 and 7 show top terms of language models of categories in a segment of the whole hierarchy extracted by the baseline and our approach. Comparing language models for topic "*music*" (at third level) extracted by the two approaches, we see that the one in Figure 6 contains too general terms like "*art*", "*entertainment*", "*news*" and "*search*" on top. On the other hand, most of the top terms in the language model extracted by our approach are strongly relevant to the topic "*music*" (in Figure 7).

At the fourth level, in Figure 6, general musical terms such as "*music*" and "*musical*" are ranked very high in the language models of categories "*musical style*", "*musical performance*" and "*musical instruments*". These terms, however, have little power to differentiate each of these categories with the others and their parents. Language model of category "*musical performance*" also contains non-relevant terms such as "*instruments*" and "*instrument*" on top. This is because training documents for this category contains noise that is about topic "*musical instruments*" instead, and the standard likelihood approach assumes all parts in the training documents are relevant. Our approach, on the other hand, exploits the relationships amongst the categories to automatically exclude non-relevant parts. As a result, the non-relevant terms do not appear on top of the language model extracted by our approach.

Similarly, at the lowest level, language models in Figure 6 contain general terms while the language models in Figure 7 focus on terms that are unique for the category at this level. Due to the space limit, we only show language models of topics in a segment of the hierarchy extracted by the two approach. But, we observed that the patterns described above hold consistently across the whole hierarchy.

[5]The preprocessed, annotated collection is available on the web site of the SYNC3 European Project: www.sync3.eu

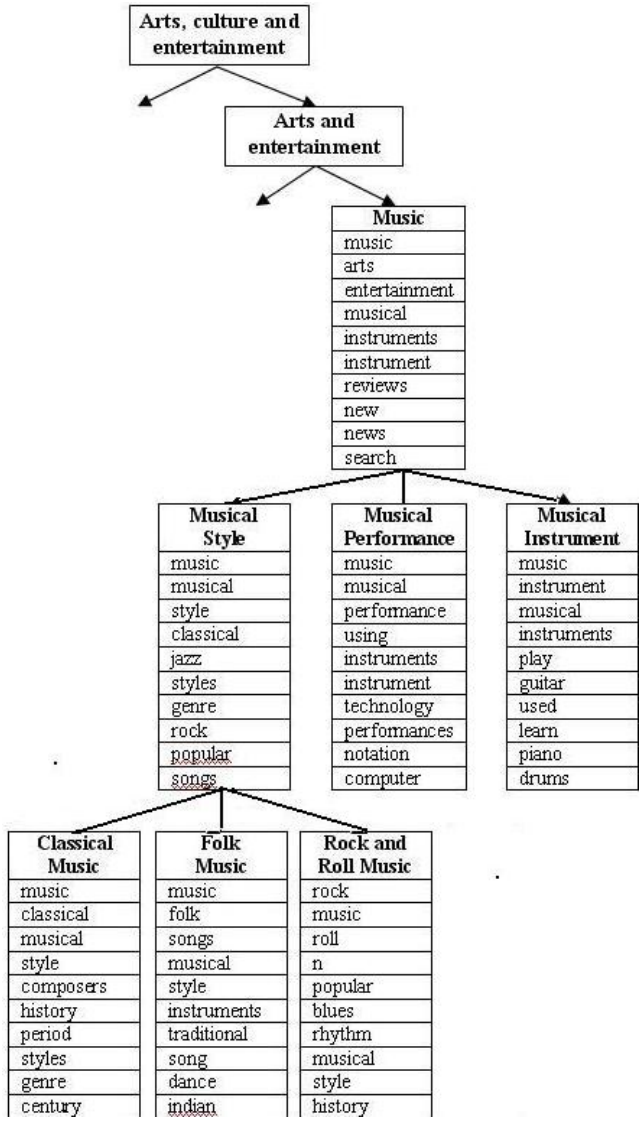

Figure 6: Topic Language Models extracted by standard maximum likelihood

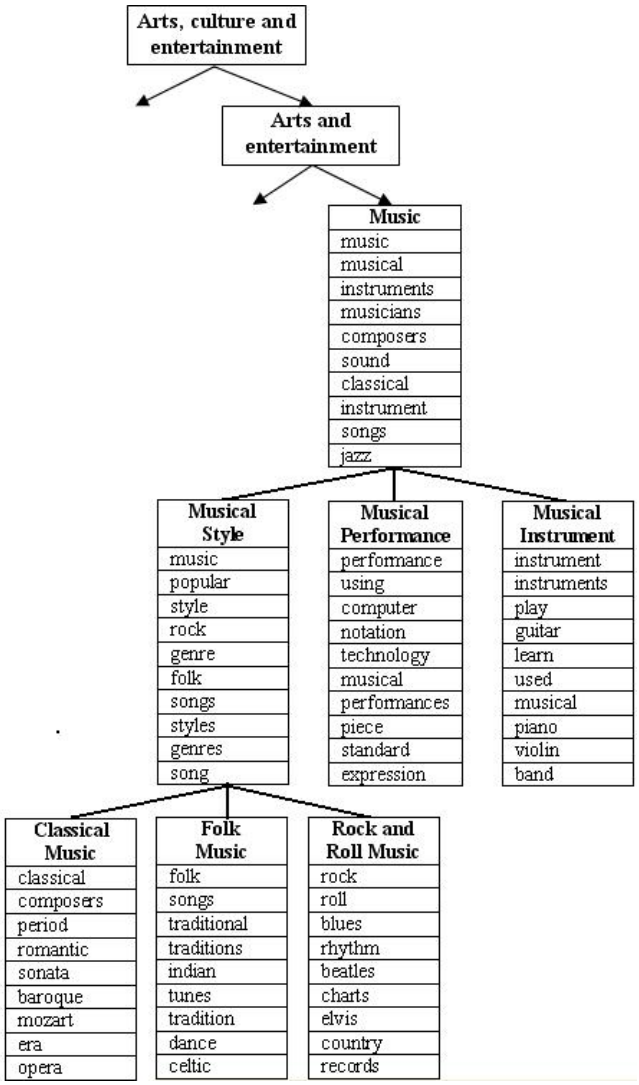

Figure 7: Topic Language Models extracted by the hierarchial topic models with ontological guidance

6.3 Classification

In this subsection, we empirically demonstrate the effectiveness of our hierarchical classification approach by comparing it with two baselines: the naive Bayes classifier and hierarchical naive Bayes classifier. We pick naive Bayes as a baseline because it has widely been shown effective for text classification, especially when training data is imperfect (Krithara et al., [16]). Hierarchical naive Bayes is an extension of naive Bayes [21]. Specifically, the language model of a category is smoothed by the language models of its ancestors (shrinkage technique).

All of the three approaches take the category language models extracted by the hierarchical topic model approach and a test document as input; they then rank the categories in decreasing order of relevance $p(category|document)$. We measure performances by precision, recall and F-1 of *top-N* categories with different values for N. N is the ranking threshold, i.e. categories ranked within top N are considered relevant to the document and the others are considered to be non-relevant. Besides standard measures of precision, recall and F-1, we also use hierarchy-based extensions of these measures as proposed in [4]. The basic idea is that it is better to classify a document into a category near the correct one in the hierarchy, than to a totally unrelated category (i.e. the cost of error depends on the similarity between the predicted category and the real ones). The similarity of two categories is defined by their respective positions in the hierarchy. We refer the readers to [4] for more details. We average the performances over all test documents.

Figure 9 shows standard precision, recall and F-1 of top-N predictions for N in range 5 to 35 by the three approaches. In terms of these standard measures, performances of the two baselines are similar. The proposed hierarchical classification approach is consistently better than naive Bayes and hierarchical naive Bayes in terms of both precision and recall. In terms of F-1, the best performance of our approach is 41%, while the best performances of naive Bayes and hierarchical naive Bayes are 16%. Note that in this large-scale text classification problem, the classifiers have to make tough decision amongst more than 1100 possible choices.

When using hierarchy-based measures (Figure 8), we could see in the figure that hierarchical naive Bayes is better than naive Bayes in terms of precision since the shrinkage smoothing technique could alleviate some imprecision in the estimation of category language models. However, the hierarchical naive Bayes is slightly worse than naive Bayes in terms of recall. This is due to the smoothing technique that makes language models of neighbour categories (i.e. categories that shares some common ancestors) highly similar. Consequently, this results in a ranked list of categories for each test document that is less diverse. As in the previous case, our approach is generally better than the two baselines in terms of both precision and recall. In terms of F-1, our approach is around 13.4% and 41.2% better than hierarchical naive Bayes and naive Bayes.

7. RELATED WORD

Our work in this study is related to several existing directions: information retrieval, document classification without labelled data, hierarchical text classification and topic modelling. We briefly review each of these directions.

The use of pseudo-positive documents as an important

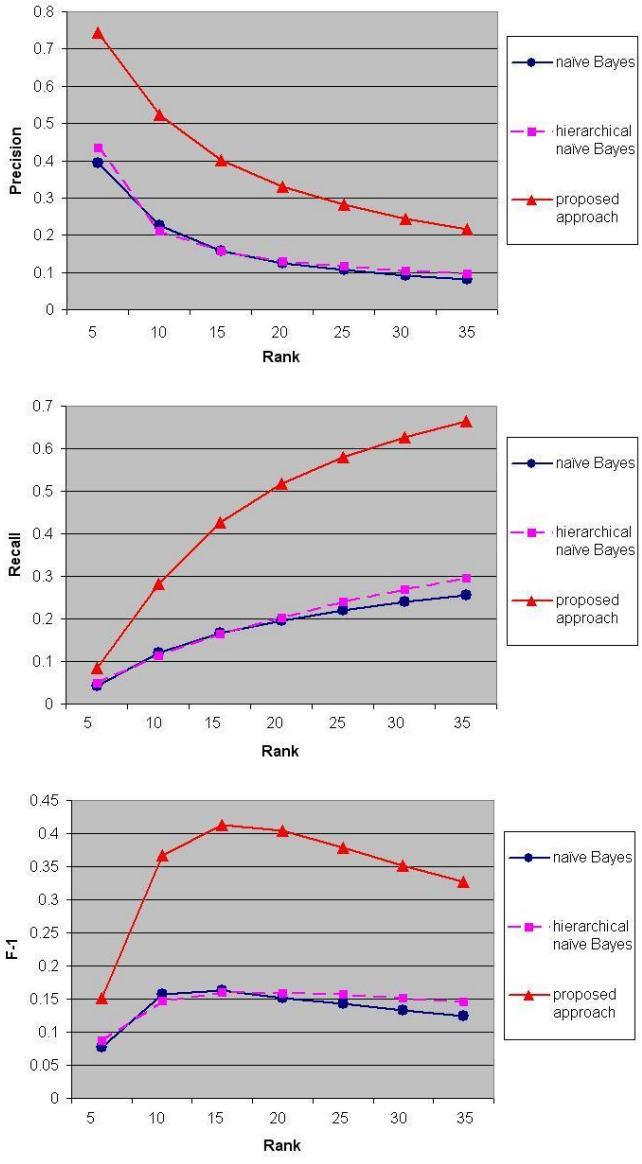

Figure 8: Classification Performances by Standard Measures. The rank corresponds to the limit on the number of predicted labels.

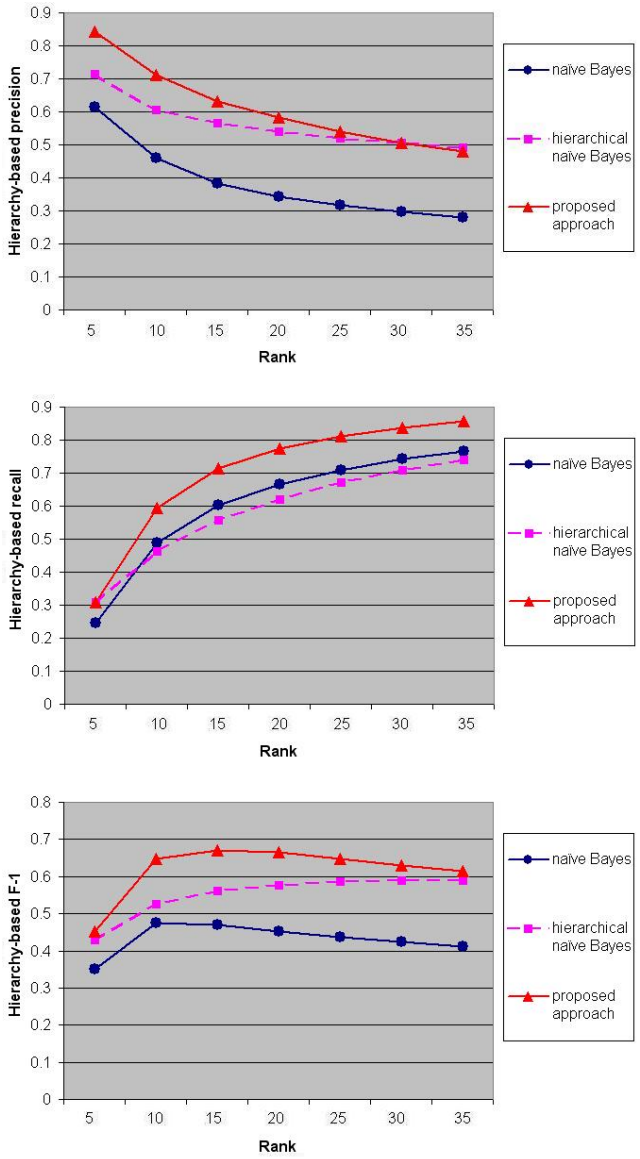

Figure 9: Classification Performances by Hierarchy-based Measures

component to bootstrap process is common in the information retrieval community. But, unlike standard ad-hoc retrieval, the most popular form of information retrieval, which aims at retrieving relevant documents for individual queries separately, our retrieval approach exploits hierarchical relationships amongst queries to improve retrieval performance.

As far as text classification without labelled data is concerned, several works have been proposed recently for building flat text classifiers without labelled data [8, 23, 25, 14, 13]. Generally, instead of using labelled documents, their approach uses retrieval or bootstrapping techniques to initially assign documents to topics represented by a title or a few keywords, then incrementally builds a classifier and refines the assignments through many iterations. This family of approaches adopts a strategy in "three phases" (initialization exploiting the prior knowledge; iterative refinement and final categorization), as our method does. However, when the topic representations are short and ambiguous, the initial assignment is likely to be inaccurate and that could mislead the whole process. Our approach proposed for hierarchical classification, on the other hand, takes into account the hierarchical relationships to automatically enrich semantic representations of topics. As a result, performance of the initial retrieval phase is improved. Second, our learning approach on initially retrieved documents is robust to noise in these documents. So, the approach could reduce the risk of depending on the initial step. Finally, in the categorization step, our approach uses information cascaded top-down (from ascendant categories) and bottom-up (from descendant categories) to alleviate any noise in each category language model estimation.

In terms of hierarchical text classification based on languages models, our work has to be related to the methods proposed in [15, 20, 6, 7]. These papers all propose supervised approaches which rely on manually labelled data. There are also several previous works exploiting hierarchical structure to improve estimated topic language models. Specifically, [18] and [21] use top-down information, while [24] use bottom-up information to smooth the estimates of $p(w|t)$ for all topics. However, in those works, the smoothing could have a side effect that makes distributions of similar topics which share common ancestors or descendants highly overlapping and less distinguishable. Our hierarchical classification approach uses both top-down and bottom-up information in an adaptive way to alleviate the problem of noise in topic language model estimations as well as to maintain the discriminative power of these language models. Moreover, our classification approach softens decisions at early stages, so it could further alleviate the issue of cascading errors.

Our approach for extracting topic language models (Phase 2) is based on a latent Dirichlet framework, which has also been widely studied in the area of purely unsupervised latent factor decomposition (or clustering), especially in "*topic modeling*" [3, 12, 10]. Hierarchical topic models are also proposed by Blei et al. [2]. Moreover, the work of [19] proposes a generative process that could explain the content of a document as generated by a topic hierarchy that is much more flexible that a standard hierarchical tree, especially by its ability to mix multiple leaves of the topic hierarchy. There is, however, a key difference between these topic models and our model. The topics discovered by these typical topic models are synthetic and do not correspond to given topics in human minds or ontological systems. However, in classification task, topics are given upfront. Our approach on the other hand, is able to generate topic language models explicitly associated to nodes within a given ontology. An interesting way of future research should be the integration of the broader, more flexible framework proposed by [19] in our method, by adapting it to our "weakly supervised" hierarchical classification setting.

8. CONCLUSIONS

In this paper, we propose a novel approach for automatic large-scale hierarchical text classification which does not require any labelled data. Instead of using human-labelled documents, we take advantage of the ontological knowledge defined in a category hierarchy to construct enriched and context-aware queries for these categories in the hierarchy and then use these queries to retrieve pseudo-relevant documents on the Web. Then, we propose a hierarchical topic model with ontological guidance, which exploits the relationships amongst categories to exclude noise, identify really relevant parts in the pseudo-relevant documents, and estimate language models for these categories. Finally, we present a novel algorithm using hierarchical structure for classifying test documents.

Our experiments on IPTC taxonomy containing 1131 categories demonstrate effectiveness of our approach. In estimating language models for categories, our experiments show that the hierarchical topic model with ontological guidance is robust to noise in pseudo-relevant documents and could be able to identify terms relevant to categories at different levels of abstraction. As a result, language models extracted by the proposed approach are more appropriate than ones extracted by the maximum likelihood. In the final phase, classifying test documents, the proposed hierarchical classification algorithm outperforms flat naive Bayes (150% and 41.2% improvement w.r.t to the standard and hierachy-based F-1, respectively) and a popular hierarchical classification approach, hierarchical naive Bayes (150% and 13.4% improvement). Overall, we show that just by taking the simple ontological knowledge defined in a category hierarchy, we could automatically build a large-scale hierarchical classifier with reasonable performance of 41.3% and 67% in terms of the standard and hierarchy-based F-1 measures.

9. ACKNOWLEDGMENTS

This work was partly funded by ICT-Content and Knowledge Program of the European Commission, under the SYNC3 Project, FP7-231854. The authors would like to thank the European Center for Journalism (EJC), especially Liliana Bounegru, for their invaluable help in annotating the news collection

10. REFERENCES

[1] H. Avancini, A. Lavelli, F. Sebastiani, and R. Zanoli. Automatic expansion of domain-specific lexicons by term categorization. *ACM Transactions on Speech and Language Processing (TSLP)*, 3(1):1–30, 2006.

[2] D. Blei, T. Griffiths, M. Jordan, and J. Tenenbaum. Hierarchical topic models and the nested chinese restaurant process. In *Advances in Neural Information Processing Systems*. MIT Press, 2004.

[3] D. Blei, A. Y. Ng, and M. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[4] E. Costa, A. Lorena, A. Carvalho, and A. Freitas. A review of performance evaluation measures for hierarchical classifiers. In *Evaluation Methods for Machine Learning II: papers from the AAAI-2007 Workshop*, pages 1–6. AAAI Press, 2007.

[5] A. A. Dayanik, D. D. Lewis, D. Madigan, V. Menkov, and A. Genkin. Constructing informative prior distributions from domain knowledge in text classification. In *Proceedings of SIGIR*, pages 493–500. ACM Press, 2006.

[6] S. Dumais. Hierarchical classification of web content. In *Proceedings of SIGIR*, pages 256–263. ACM Press, 2000.

[7] E. Gaussier, C. Goutte, K. Popat, and F. Chen. A hierarchical model for clustering and categorising documents. In *Proceedings of ECIR*, 2002.

[8] A. Gliozzo, C. Strapparava, and I. Dagan. Improving text categorization bootstrapping via unsupervised learning. *ACM Transactions on Speech and Language Processing (TSLP)*, 6(1), 2009.

[9] S. Godbole, A. Harpale, S. Sarawagi, and S. Chakrabarti. Document classification through interactive supervision of document and term labels. In *Proceedings of PKDD*, pages 185–196, 2004.

[10] T. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101:5228–5235, 2004.

[11] V. Ha-Thuc, Y. Mejova, C. Harris, and P. Srinivasan. News event modeling and tracking in the social web with ontological guidance. In *Proceedings of IEEE International Conference on Semantic Computing*, 2010.

[12] T. Hofmann. Probabilistic latent semantic analysis. In *Proceedings of Uncertainty in Artificial Intelligence, 1999*, pages 289–296, 1999.

[13] C.-M. Hung and L.-F. Chien. Web-based text classification in the absence of manually labeled training documents. *JASIST*, 58(1):88–96, 2007.

[14] Y. Ko and J. Seo. Learning with unlabeled data for text categorization using bootstrapping and feature projection techniques. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, 2004.

[15] D. Koller and M. Sahami. Hierarchically classifying documents using very few words. In *Proceedings of ICML*, 1997.

[16] A. Krithara, M. Amini, J. michel Renders, and C. Goutte. Semi-supervised document classification with a mislabeling error model. In *Proceedings of ECIR*, 2008.

[17] A. Mccallum and K. Nigam. Text classification by bootstrapping with keywords, em and shrinkage. In *Workshop for Unsupervised Learning in Natural Language Processing*, pages 52–58, 1999.

[18] A. Mccallum, R. Rosenfeld, T. Mitchell, and A. Ng. Improving text classification by shrinkage in a hierarchy of classes, 1998.

[19] D. Mimno, W. Li, and A. McCallum. Mixtures of hierarchical topics with pachinko allocation. In *Proceedings of ICML*, pages 633–640. ACM, 2007.

[20] A. Sun and E.-P. Lim. Hierarchical text classification and evaluation. In *Proceedings of ICDM*, 2001.

[21] K. Toutanova and F. Chen. Text classification in a hierarchical mixture model for small training sets. In *Proceedings of CIKM*, pages 105–113. ACM Press, 2001.

[22] S. Veeramachaneni, D. Sona, and P. Avesani. Hierarchical dirichlet model for document classification. In *Proceedings of ICML*, volume 119, pages 928–935. ACM, 2005.

[23] P. Wang and C. Domeniconi. Towards a universal text classifier: Transfer learning using encyclopedic knowledge. In *Proceedings of ICDM Workshops*, 2009.

[24] R. Wetzker, T. Alpcan, C. Bauckhage, W. Umbrath, and S. Albayrak. An unsupervised hierarchical approach to document categorization. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, pages 52–58, 2007.

[25] C. Zhang, G.-R. Xue, and Y. Yu. Knowledge supervised text classification with no labeled documents. In *Proceedings of PRICAI*. Springer, 2008.