Classification of Brand Names Based on n-Grams

Pattaraporn Warintarawej, Anne Laurent, Pierre Pompidor, Benedicte Laurent

Classification of Brand Names Based on n-grams

P. Warintarawej, A. Laurent, P. Pompidor
LIRMM - Univ. Montpellier 2
CNRS UMR 5506 - Montpellier
France
{*warintarawej,laurent,pompidor*}*@lirmm.fr*

B. Laurent
Namae Concept
Montpellier
France
b.laurent@namaeconcept.com

***Abstract*—Supervised classification has been extensively addressed in the literature as it has many applications, especially for text categorization or web content mining where data are organized through a hierarchy. On the other hand, the automatic analysis of brand names can be viewed as a special case of text management, although such names are very different from classical data. They are indeed often neologisms, and cannot be easily managed by existing NLP tools. In our framework, we aim at automatically analyzing such names and at determining to which extent they are related to some concepts that are hierarchically organized. The system is based on the use of character n-grams. The targeted system is meant to help, for instance, to automatically determine whether a name sounds like being related to *ecology*.**

***Keywords*-Brand Names; Textual Classification; n-grams; Hierarchies;**

I. Introduction

Brand names are very important as they occur everywhere in our lives and allow companies and organizations to associate distinctive signs on themselves, and their products, services. However, they cost a lot of money to companies if they have to be changed due to bad choices. The challenge is thus to provide automatic tools to analyze such names. These names are often neologisms and must be analyzed in many contexts (languages, social classes, etc.).

In this work, the goal is to analyze brand names so that the end user can be informed about the class which the names is the closest to. For instance, one name could be very close to the concept of *ecology*. The distance we use is based on a lexical analysis based on sequences of characters (n-grams). Further work will help classifying the names depending on their semantic or phonetic distances. In the currently implemented system, the user can analyze a name he has just created. The input is a name (existing or not in dictionaries) proposed by the user, and the system then shows him/her which concepts are related to this name. For this purpose, starting from this input, the system checks which concepts share the most important number of representative n-grams.

For this purpose, we consider the concepts from the Larousse Thesaurus [1], which is in French and is equivalent to the Roget Thesaurus [2]. In many applications and domains, the concepts are organized into a hierarchy. As shown on Figure 1, this is the case for the thesaurus being considered here. The 873 lowest concepts are organized into 95 themes which are regrouped into 27 sections, and finally into 3 classes.

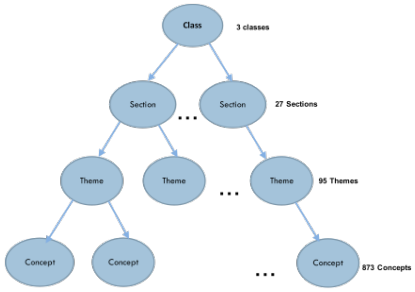


Figure 1: Larousse Thesaurus

In this paper, we study how to classify brand names into these hierarchically organized concepts. In the first part, we recall why brand names are difficult to manage and the existing approaches for hierarchical classification, before presenting our method and reporting experimental results.

II. Related Work

A. Brand Names and Concepts

Brand names are very important in order to identify the products, services or companies they refer to. It is thus crucial to provide people working on trademarks (e.g. in marketing companies) with tools that allow to automatically analyze what brand names may evoke. However, as reported in [3], brand names have been hardly studied in linguistics, although they are an interesting kind of nouns (and expressions). Much research is indeed being done in economics and law regarding trade names used in advertising (which we would refer to as advertising names). However few works are published on the subject in French linguistics. These names can be either taken from a dictionary, or be neologisms. They have been often associated with proper names. However, they must be studied as a very particular piece of language.

B. Hierarchical Classification

Supervised classification (and text classification) has been studied for many years [4], [5] and is a well-known technique to associate objects to predefined categories. In this work, we address a particular problem of supervised learning, as the predefined categories are organized into a hierarchy.

Many works are based on n-grams. An n-gram is a subsequence of n items from a given sequence. These items can for instance be phonemes, syllables, letters or words. Stopword lists are often used to clean data. A stopword list is a set of words that occur frequently in the language and which are not discriminant for the categories. In the other word, they are not meaningful. For instance, the word *the* in English does not convey any idea of a related concept.

[6] introduces a simple method using n-grams based on characters. The authors assume that documents from the same category should share similar n-grams. The training data set is taken from from newsgroup and is split into n-grams (1-5) that are counted. The top 300 highest frequency n-grams are selected to be the feature set. The authors use the simple rank-order statistic called out-of-place measure to compare new documents and categories. They then pick the category having the smallest distance. No model is learnt.

In [7], the author also uses n-grams. The approach mixes the ideas of out-of-place measure and the cosine similarity measure from vector space model. This paper proposes two ways of feature selection to represent the document, namely the inverse document frequency (idf) by selecting features greater than a threshold and the top 1,000 n-grams having the highest frequency. The result shows that top 1,000 n-grams with highest frequency is more successful than idf.

[8] proposes an approach for classifying Web documents into large topic ontology. The training data set comes from DMoz topic ontologies that contains web pages organized into a hierarchy consisting of 15 levels with 94,113 non-leaf nodes. Data is pre-processed for removing standard English stop-words (525) and stemming the words using the Porter stemmer [9]. Bag of words are used for representating the information, by computing TF-IDF as feature vector. The k-nearest neighbor classifier is used for classifying a new document. In classifying phase, all the topics from the hierarchy are considered as a flat structure by building separate classifiers for every class in the hierarchy.

[10] is one of the closest work compared to our approach as the authors use thresholds to classify a document into a hierarchical structure. The authors use a top-down approach, meaning that the classification starts from top level and filters all categories in top level by threshold. The hierarchical structure from LookSmarts web directory is used. For the non-hierarchical case, the authors select 1,000 terms (words) from each of 150 categories by the largest mutual information between a test document and a category. For the hierarchical case, the authors also select 1,000 features from each of 13 top-level categories and 1,000 features from each of 150 second-level categories. During the training phase and classifying, SVM are used as a classifier. In the classifying phase, scores are combined from top and second level models using different combination rules.

In [11], the objective is to classify documents into appropriate classes by taking advantage of a hierarchy of classes. The authors use the naïve Bayes classifier and propose a new statistical technique called *shrinkage* to improve the estimation of the parameters. The idea of shrinkage is to smooth the parameter estimation of a node by interpolating all its parent modes. The feature selection for classes is performed by ranking the highest mutual information at each internal node of the hierarchy. The authors claim that the shrinkage method can reduce text classification error by up to 29%, especially when the training data is sparse and the number of classes is large.

In [12], the authors propose an approach that utilizes the hierarchical structure by focusing a very small set of features. Using the Reuters-22173 dataset, they define top-level categories by merging sub-categories (e.g. Corn and Wheat group into Grain or Dollar and Interest Rate group into Money Effect). For classification, the naïve Bayes classifier is used, with a top-down approach. The test documents are classified into top-level first and then filtered to the first-level categories. The result when comparing between flat model and hierarchical models demonstrates the advantage of hierarchical model with a small number of features.

In [13], the authors address the problem of protein classification, using n-grams as the descriptors of their objects. These n-grams are mined using association rule-based methods and SVM are used for the classification task.

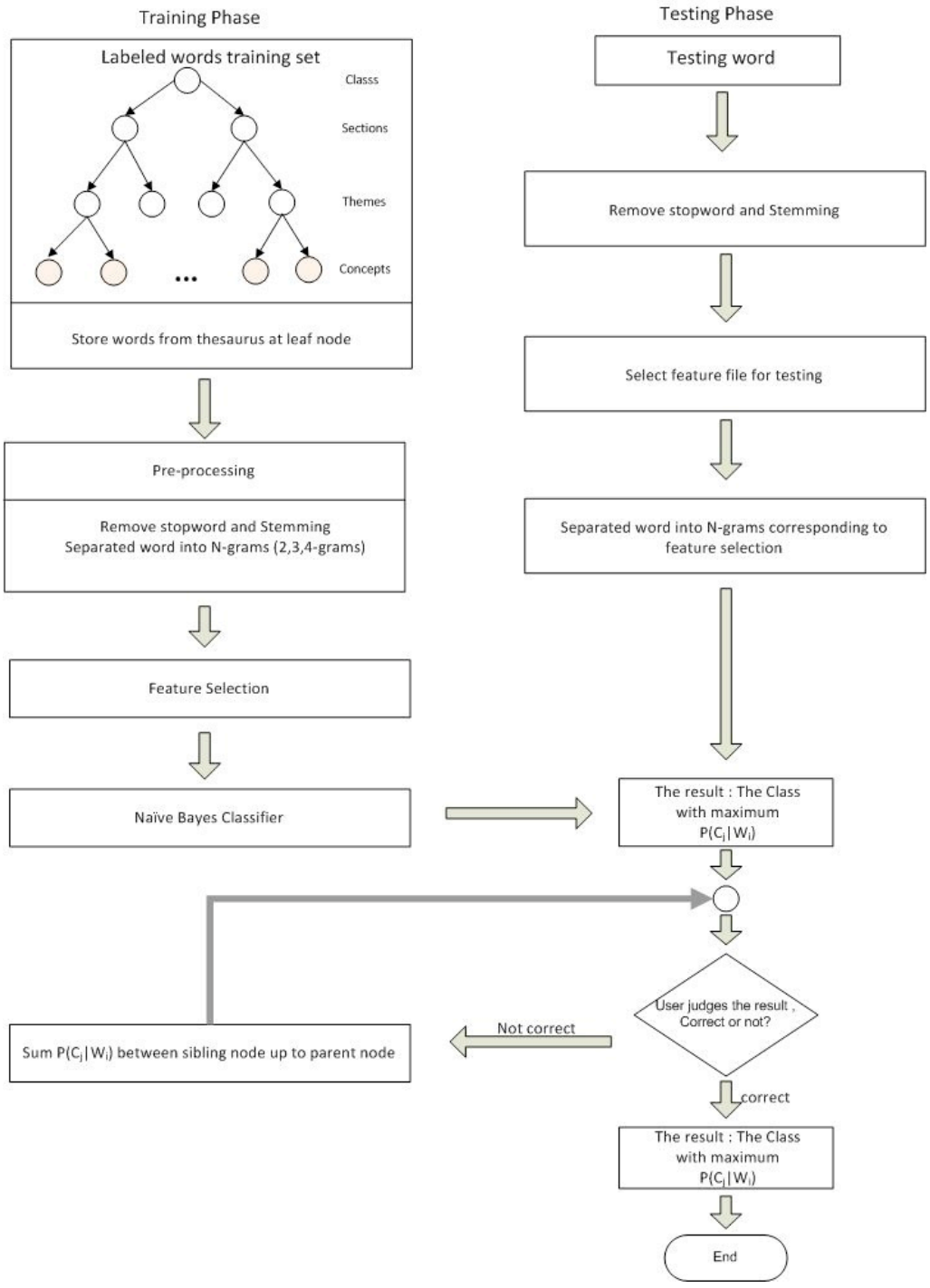
III. HCBN: Hierarchical Classification of Brand Names

The general process we propose is described in Figure 2.

The methodology for building the model for hierarchical classification of brand names is split into the following steps: (1) Preparing Data, (2) Collecting words into leaf node from French Larousse, (3) Applying Stopword and Stemming (4) Split word into n-grams (2-grams, 3-grams, 4-grams), (5) Selecting the most relevant features.

For classifying a new brand name, the system (1) first gets the new name from the user, (2) then splits the name into n-grams (2-grams, 3-grams, 4-grams), and (3) uses a naive Bayes classifier to classify at leaf node. (5) If the user judges the result as being not satisfactory, then the system tries to sum up the probabilities of siblings to roll up to a higher level in the class hierarchy. It then reports the class that gives the maximum value.

It should be noted that this last step is the main important novel part of our method. We detail the whole approach below.


Training Phase
Labeled words training set
Classs
Sections
Themes
Concepts
Store words from thesaurus at leaf node
Pre-processing
Remove stopword and Stemming
Separated word into N-grams (2,3,4-grams)
Feature Selection
Naïve Bayes Classifier
Testing Phase
Testing word
Remove stopword and Stemming
Select feature file for testing
Separated word into N-grams corresponding to feature selection
The result : The Class with maximum P(Cj|Wi)
User judges the result , Correct or not?
Not correct
Sum P(Cj|Wi) between sibling node up to parent node
correct
The result : The Class with maximum P(Cj|Wi)
End

A. Building Information about the Concepts

In existing text classification methods, it is important, for every concept, to gather related words. These words are then split into n-grams in order to compute the most discriminant n-grams. In our case, the challenge is to choose how to associate discriminant n-grams at several levels of granularity.

Two alternatives are possible: either to gather related names for every entry from the hierarchy (concepts, themes, sections, classes), or to gather related names only at the lowest level (concepts) and aggregate them for building discriminant n-grams at the upper levels (themes, sections, classes). In this work, we choose to gather words at the leaf level.

The data we gather is first cleaned. For this, we remove common words (stopwords). We use a French stopword list from the snowball project [14]. After that we use the Porter stemming algorithm to transform words into their grammatical roots. The stemming is the process for reducing inflected (or sometimes derived) words to their stem, base or root form.

For our task we consider n-grams of characters (letters, punctuation marks), assuming that words belonging to the same category should have some common sequences of letters. We separate a training set of words and a test set, and we choose 2-grams, 3-grams and 4-grams to be the feature set of the classification model.

The main characteristic of text classification is the high dimensionality of textual data. Usually in text classification, a document is represented by using a bag of words. In this work task, we use characters n-grams to be the feature set, and the bag of word is represented by the sequences of characters vector corresponding to a given word, thus leading to a very high number of attributes to be managed. Methods have been proposed to select the most relevant attributes. These methods, also known as feature selection, allow to reduce the dimensionality of the data, and to improve classification effectiveness and computation efficiency. As described in [15], several methods exist. Statistics (chi-square) and information gain are some of the more effective ones for optimizing classification result. In this work, we apply and compare the results obtained using several feature selection methods.

B. Classifying over Hierarchical Concepts

The naïve Bayes classifier is a simple classifier model that base on probabilistic theory call Bayesian theorem with independence assumptions so-called naïve Bayes assumption. All attributes are assumed to be independent, which is in fact not correct in real-world text classification. However, naïve Bayes performs well [16].

To classify a document into a class, the best class is selected by maximum a posterior probability (MAP). We use naïve Bayes assumption, all of words in a document are independent and also position of words in the document. To apply naïve Bayes classifier to classify words into a class, we consider the classes from the hierarchical word structure derived from the French Larousse thesaurus. We consider a word as a vector space that consists of sequences of consecutive characters.

We assume that a word is composed of a sequence of consecutive characters called n-grams. For obeying naïve Bayes assumption, we also assume that these sequences are independent. The main idea behind our task is that a word will belong to a class, because it has some sequences of consecutive characters matching with other words from the same class. We thus separate all words into n-grams which become our attributes. If a word contains any such attribute (i.e. sequence), we mark 1 to be a weight of vector and 0 otherwise. Thus a word is represented by binary vector. We use multi-variate Bernoulli model.

Although the result of the comparison between muti-variate Bernoulli model and multinomial model is shown that the multinomial performs usually better at larger vocabulary sizes than the muti-variate Bernoulli model but the muti-variate Bernoulli model works well with small vocabulary size [17].

For using n-grams sequences to be a feature, we only use a sequence absence or presence in word and we do not consider the frequency of consecutive characters sequences occurring in a word. After classifying at leaf node by naïve classifier, we try to improve classifying performance by using hierarchical structure of words.We purposed new methodology of hierarchical classification based on independence assumption of naïve bayes theory.

When classification is not satisfactory at the leaf level (i.e., the probability is lower than a predifined threshold), then we use the relationship between sibling nodes to go up in the hierarchy. For this purpose, we sum up the probabilities of all children to compute the score of the upper level node. At this upper level, the category having the highest score is chosen as the class to be output to the user. If this class is not satisfactory (i.e. the classification score is lower than the predefined threshold), then this process is repeated by rolling up to the upper level as many times as possible and necessary.

IV. Experimental Results

Words have been collected using an Internet search related to each concept from the French Larousse thesaurus. The hierarchical structure of the data consists of 4 levels: 3 Classes, 27 Sections, 95 Themes and 873 Concepts from top to leaf level. 53,778 words have been collected at leaf node. In our experiment, we selected 6 concepts from different classes containing 821 words. From 821 words, we separate them into training set and testing set by random selection (see Figure 3).

Concept Name	Number of words	
	Training set	Testing set
Existence	106	45
Matérialité	44	19
Homme	45	20
Femme	89	38
Vêtement	281	121
Chaussure	9	4
Total	547	247

Figure 3: Corpus used in the Experiments (in French)

We start classifying at leaf node level by considering 6 concepts consisting of 821 words. These words have been split into 547 words for training set and 247 for testing set. Words from training set are separated into n-grams (401, 1,735 and 3,075 sequences for 2-gram, 3-grams and 4-grams respectively). Before building a bag of words, feature selection techniques are used to select a set of discriminating features. Three methods were evaluated, including TopRank frequency, chi-square (statistics), Information Gain. For 2-grams, we select all the features because the total of 2-grams sequence is 401.

We run naïve classifier (multi-variate bernoulli model) and compare the results depending on the feature selection methods and the number and the size of features.

A. *Results*

Figures 4 and 5 show classification accuracy and macro and micro-averaged precision and recall using three types of feature selection techniques with various n-grams and feature size.

Firstly, for overall result of classification accuracy, 2-grams with 401 features achieved the best accuracy at 67.61%. Secondly, when comparing between several feature sizes (300, 500, 1000), 300 features of 3-grams with chi-square achieved the highest accuracy score at 57.49%. The second accuracy score is given by 3-grams with 500 features and information Gain at 57.09%.

For every type of feature selection technique, using 1,000 features performed poorly (except for TopRank frequency) even if [5], [11] claim that a larger vocabulary generally performs better than small sizes. This may be due to the use of Bernoulli Model.

Thirdly, the results show that 2-grams perform better than 3-grams and 4-grams. However for 2-grams, the number of features is too small in our experiment so we cannot compare 2-grams between feature selection types and feature sizes.

n-grams	Feature Type	#No of Feature	Correctly Classified (%)
2-grams	TopRank	401	**67.61**
3-grams	ChiSquare	300	**57.49**
3-grams	ChiSquare	500	56.68
3-grams	ChiSquare	1000	55.47
3-grams	InfoGain	300	56.68
3-grams	InfoGain	500	56.68
3-grams	InfoGain	1000	55.47
3-grams	TopRank	300	55.47
3-grams	TopRank	500	**57.09**
3-grams	TopRank	1000	55.87
4-grams	ChiSquare	300	56.28
4-grams	ChiSquare	500	53.85
4-grams	ChiSquare	1000	52.63
4-grams	InfoGain	300	56.68
4-grams	InfoGain	500	53.85
4-grams	InfoGain	1000	52.63
4-grams	TopRank	300	54.25
4-grams	TopRank	500	53.85
4-grams	TopRank	1000	53.04

Figure 4: Experimental Results: Classification Accuracy on three Types of Feature Selection Techniques with Various n-grams and feature size

n-grams	Feature Type	#No of Feature	Macro-averaged			Micro-averaged
			Precision	Recall	F-Measure	Precision/Recall/F-Measure
2-grams	TopRank	401	0.4964	**0.4328**	**0.4437**	**0.6761**
3-grams	ChiSquare	300	**0.5489**	**0.2765**	**0.2874**	**0.5749**
3-grams	ChiSquare	500	0.5069	0.2495	0.2431	0.5668
3-grams	ChiSquare	1000	0.3152	0.2310	0.2146	0.5547
3-grams	InfoGain	300	0.4495	0.2546	0.2533	0.5668
3-grams	InfoGain	500	0.5025	0.2495	0.2423	0.5668
3-grams	InfoGain	1000	0.3152	0.2310	0.2146	0.5547
3-grams	TopRank	300	0.4876	0.2467	0.2420	0.5547
3-grams	TopRank	500	0.5011	0.2576	0.2550	**0.5709**
3-grams	TopRank	1000	0.3430	0.2354	0.2222	0.5587
4-grams	ChiSquare	300	0.5021	0.2469	0.2484	0.5628
4-grams	ChiSquare	500	0.5495	0.2189	0.2056	0.5385
4-grams	ChiSquare	1000	0.3976	0.2014	0.1734	0.5263
4-grams	InfoGain	300	0.6593	**0.2552**	**0.2636**	0.5668
4-grams	InfoGain	500	0.5166	0.2189	0.2051	0.5385
4-grams	InfoGain	1000	0.3976	0.2014	0.1734	0.5263
4-grams	TopRank	300	**0.6614**	0.2316	0.2284	0.5425
4-grams	TopRank	500	0.3616	0.2132	0.1923	0.5385
4-grams	TopRank	1000	0.3980	0.2058	0.1814	0.5304

Figure 5: Macro and Micro-averaged precision recall and F-measure results from three type of feature selection techniques with various n-grams and number of features

We can only compare between 3-grams and 4-gram. In this case, the 3-grams perform better than 4-grams.

Regarding the types of feature selection, chi-square achieves better than information gain and TopRank on average.

The results show that the highest micro-averaged performance is achieved with 2-grams with TopRank at 401 features. When comparing feature selection techniques on 3-grams and 4-grams, chi-square with 3-grams and 300 features and TopRank frequency with 3-grams and 500 features give the best classification evaluation measures by micro-averaged f-measure.

When comparing the results depending on the feature size, we show that chi-square with 3-grams and 300 features gives the best classification evaluation measure by micro-averaged f-measure. The second is given by TopRank frequency with 3-grams and 500 features. From these results, we can conclude that when the size of feature increases, then the performance slightly decreases.

V. Conclusion

This paper presents a system devoted to automatic brand name classification, with applications over hierarchically-organized classes. The solution proposed here consists in using representative character n-grams and defining a strategy to aggregate them over the hierarchy. Experiments have been carried out on the French classification of concepts from the *Larousse*. The experimental results show the interest of our approach.

The future research directions mostly include the extension of experiments, and the study of classification methods based on phonems [18]. Moreover, we will study how to improve our system in order to support multi-linguism [19].

References

[1] D. Pechoin, *Thesaurus - des mots aux idées, des idées aux mots*. Larousse, 1991.

[2] P. Roget, *Roget's Thesaurus of English Words and Phrases*, 2004.

[3] A. Laurent, B. Laurent, D. Brouillet, S. Martin, and M. Roche, "Embedding emotions within automatically generated brand names," in *Proc. of the Int. Conference on Kansei Engineering and Emotional Research (KEER'2010)*, 2010.

[4] T. M. Mitchell, *Machine Learning*. New York: McGraw-Hill, 1997.

[5] T. Joachims, *Learning to Classify Text Using Support Vector Machines – Methods, Theory, and Algorithms*. Kluwer/Springer, 2002.

[6] W. B. Cavnar, "Using an n-gram-based document representation with a vector processing retrieval model," in *TREC*, 1994, pp. 0–.

[7] P. Nather, "N-gram based text categorization," Master's thesis, Univerzita Komenskho - Fakulta matematiky, fyziky a informatiky - Katedra aplikovanej informatiky, 2005.

[8] M. Grobelnik and D. Mladenic, "Simple classification into large topic ontology of web documents," *CIT*, vol. 13, no. 4, pp. 279–285, 2005.

[9] M. F. Porter, "An algorithm for suffix stripping," *Program*, vol. 14, pp. 130–137, 1980.

[10] S. T. Dumais and H. Chen, "Hierarchical classification of web content," in *SIGIR*, 2000, pp. 256–263.

[11] A. McCallum, R. Rosenfeld, T. M. Mitchell, and A. Y. Ng, "Improving text classification by shrinkage in a hierarchy of classes," in *Proc. of the int. conf. on Machine Learning (ICML)*, 1998, pp. 359–367.

[12] D. Koller and M. Sahami, "Hierarchically classifying documents using very few words," in *Proc. of the int. conf. on Machine Learning (ICML)*, 1997, pp. 170–178.

[13] F. Mhamdi, R. Rakotomalala, and M. Elloumi, "A hierarchical n-grams extraction approach for classification problem," in *Int. Conf. on Signal-Image Technology and Internet Based Systems (SITIS)*, 2006.

[14] S. Project. [Online]. Available: http://snowball.sourceforge.net

[15] Y. Yang and J. O. Pedersen, "A comparative study on feature selection in text categorization," in *Proceedings of the Fourteenth International Conference on Machine Learning*. Morgan Kaufmann Publishers, 1997, pp. 412–420.

[16] P. Domingos and M. Pazzani, "On the optimality of the simple bayesian classifier under zero-one loss," *Machine Learning*, vol. 29, 1997.

[17] A. McCallum and K. Nigam, "A comparison of event models for naive bayes text classification," in *In AAAI/ICML-98 Workshop on Learning for Text Categorization*. AAAI Press, 1998, pp. 41–48.

[18] C. Corredor-Ardoy, P. B. de Mareüil, M. Adda-Decker, L. Lamel, and J. Gauvain, "Classement automatique de phonèmes dans un cadre multilingue," in *In Proc. XXIIimes Journées d'Etudes sur la Parole*, 1998, pp. 75–78.

[19] P. P. K. Chowdary, "MLPOSTGRES: Implementing Multilingual Functionalities Inside PostGreSQL Database Engine," Master's thesis, Indian Institute of Science, Bangalore, India, June 2005.