Why Text Segment Classification Based on Part of Speech Feature Selection

Iulia Nagy[*], Katsuyuki Tanaka, and Yasuo Ariki

Kobe University
1-1 Rokkodai-cho, Nada-ku, Kobe 657-8501, Japan
{nagy,katsutanaka}@me.cs.scitec.kobe-u.ac.jp, ariki@kobe-u.ac.jp

Abstract. The aim of our research is to develop a scalable automatic why question answering system for English based on supervised method that uses part of speech analysis. The prior approach consisted in building a why-classifier using function words. This paper investigates the performance of combining supervised data mining methods with various feature selection strategies in order to obtain a more accurate why classifier.Feature selection was performed a priori on the dataset to extract representative verbs and/or nouns and avoid the dimensionality curse. LogitBoost and SVM were used for the classification process. Three methods of extending the initial "function words only" approach, to handle context-dependent features, are proposed and experimentally evaluated on various datasets. The first considers function words and context-independent adverbs; the second incorporates selected lemmatized verbs; the third contains selected lemmatized verbs & nouns. Experiments on web-extracted datasets showed that all methods performed better than the baseline, with slightly more reliable results for the third one.

Keywords: Question-answering, supervised learning, feature selection.

1 Introduction

In the past years Internet has become a major source of information, many people relying on it to find the answers to their questions. Although very popular, search engines do not provide the user with a direct answer to his or her query but with a number of web pages the user has to browse manually to obtain the information he or she is looking for. A crucial step for the next generation search engines is to integrate a system allowing the user to obtain a straightforward and concise answer to his or her question. Such systems are known as question-answering (QA) systems and have undergone significant progress during past years. Two main types of question-answering systems can be distinguished : factoid, which address questions requiring simple answers such as person name, organization name, numeric expression, and non-factoid dealing with questions that require a more complex answer.

[*] Exchange student from INSA de Lyon, Computer Science Department.

B. Pfahringer, G. Holmes, and A. Hoffmann (Eds.): DS 2010, LNAI 6332, pp. 87–101, 2010.

Our work focuses on creating a QA system for non factoid questions, more precisely a why-type QA system. While many such systems are presented in the QA literature, some of them suffer from domain dependency, since they address a specific domain such as medicine, or may prove difficult to build due to handcrafted patterns and the considerable grammar expert knowledge needed. In the attempt to overcome these flaws, we adopted a machine learning approach for building our why-type QA system. The main purpose of our research is to build an effective QA system able to detect why text segments from arbitrarily built corpora and scalable to different languages.

More specifically, the task we address is building a classifier for QA-system able to identify the answers that actually respond to a why question. By applying this classifier in a preprocessing step we should be able to reduce the amount of data to analyze, by eliminating all text segment not answering a why question, and therefore facilitate the work of the answer extraction module of the QA system. Previous work focused on adapting to English an approach described in the Japanese literature [10] and evaluating its performance. In this method only function words are extracted from pre-labeled text segments, and then used to train a why-classifier. Considering the overall satisfying results of this experiment, we have decided to seek for methods to improve the performance of the existent classifier.

In this paper, we present the different techniques we applied in order to improve the initial classifier's performance. In order to achieve our goal we decided to enrich the initial feature space with other valuable features. Initially we added context-independent[1] adverbs to the feature space that contained only function words. Afterwards, using a priori feature selection techniques, lemmatized verbs and lemmatized verbs & nouns were also added to the feature space.

In order to evaluate how well our 3 methods work, we trained classifiers using both LogitBoost and SVM with a Pearson VII function based kernel. Moreover, in order to ensure the validity of our experience, we used various training and evaluation datasets, composed of web-extracted text segments.

This article is organized as follows: Section 2 describes the related work on why-type QA, Section 3 describes the previous work along with the method that initially inspired us. Section 4 presents the feature selection algorithms and the classifier algorithms proposed while Section 5 describes the experimental preparation and the results. Finally Section 6 presents the conclusion and the description of future works.

2 Related Work

With the continuous growth of the information base available on Internet, the importance of effective question-answering tools to facilitate the search process continues to increase. While research in building factoid QA systems has a long

[1] Context-independent words refer to words that have no intrinsic meaning; on the contrary, context-dependent words describe an action, a feeling or an object.

history, it is only recently that studies have started to focus also on the creation and development of QA systems for answering why-type questions.

One of the best known figures in the domain is Verberne [13–16] whose initial work consisted in retrieving why-answers with the use of Rhetorical Structure Theory. In [15] she presented a re-ranking method where the score assigned to a QA-pair by QAP ranking algorithm[2] is weighted by taking into consideration a number of syntactic features. In her latest work [16] Verberne implements a fully functional why-QA system by integrating the re-ranking algorithm described in paper number and also makes a throughout analysis of the advantages and disadvantages of the BOW model in a why-QA context. This system obtains a 20% improvement in terms of MRR. Though efficient this method is labor intensive: the values produced by the 2 parsers used, the Pelican (constituency parser) and the EP4IR parser (statistical parser), have to be extracted manually and assigned to the selected features. Moreover, this method requires advanced language processing skills that only an expert in language syntax and semantics would possess.

A slightly different approach encountered in scientific literature is to derive causal expression patterns by extracting causal expressions from corpora. More clearly, these methods extract why-answers based on the presence of certain causal verbs [4] or relators [2] in the text analyzed. Although they are simple to implement and effective, these methods have the disadvantage of a low domain coverage: they do not address all why-type QA but only those that fulfill a certain pattern.

A more general approach, where causal expressions are acquired automatically with the aid of the Japanese EDR[3] dictionary, is described by Higashinaka and Isozaki [5]. The EDR dictionary contains phrases gathered from heterogeneous sources thus a good coverage of causal expressions is ensured. In this approach each phrase of the EDR dictionary is processed and context-independent words that express cause are extracted. All other words are replaced with a "*" to maintain the structure of the phrase. The structures obtained, combined with manually extracted causality indicative rules, are used to train a ranker. While known to be the best-performing fully implemented why-QA system for Japanese, Higashinaka and Isozaki's system relies on information extracted from a hand-crafted resource and therefore is not fully automated. Moreover the EDR dictionary is a rather high-priced resource only available for a limited number of languages.

To overcome the disadvantages of the former method, Tanaka [9, 10] built a fully automated classifier using bag-of-words features. Although the classifier performed well on small datasets, it failed on very large ones. In order to improve the performance of his initial method, Tanaka removed all context-dependent terms (e.g. nouns, verbs, adjectives etc.) and only included in the analysis a small group of words: the function words. Since the dimension of the new feature space

[2] QAP is a scoring algorithm for passages developed for question answering tasks. For further detail refer to [15].

[3] Electronic Dictionary Research.

was rather small the dimensionality problem was corrected, while all the initial qualities of the system were preserved. The latter method has the advantage of being easy to implement, scalable and effective. Moreover, it proves that feature selection is a promising technique in classifying text samples. Therefore our previous work was dedicated to testing and adapting it to English.

3 Previous Work

In this section we document our efforts [7] to extend Tanaka's [10] method to English. A detailed description is needed because this paper presents our attempts to improve this method.

3.1 Terminology

A content word refers to a word that has a meaning, and usually serves to describe an action, a feeling, an object (e.g. verb, noun, adjective etc.).

A function word is defined as a word that holds no meaning in itself, its sole purpose being to connect and create relations between content words.

A text segment is a group of sentences that are an eligible candidate for answering a why-question.

Tanaka's [10] method will be referred to as "Bag of function words" henceforth.

Text segments that are eligible why-answers will be referred as why-TS while those that do not as other-TS.

3.2 Bag of Function Words - Method Outline

The fundamental quality of this method is its ability to build domain independent fully automated classifiers. In his work Tanaka argues that 3 conditions are primordial to obtaining the domain independence of a classifier:

- convergence and reasonable size of feature space
- generality of features in the feature space
- ability of the feature to discriminate between encoding or not encoding causation text segments.

After analyzing vocabulary syntax, Tanaka concluded that function words fulfill all three conditions stated beforehand: their number is limited contrary to words like nouns; they have no intrinsic meaning therefore they ensure generality of features; and, last but not least, each one of them can be used to express a specific context(definition, cause, explanation etc.).

In order to identify function words in corpora, Tanaka used syntactic parser for Japanese on each text segment. The words that fulfilled the conditions stated above were selected and included in the feature space; the subset obtained contained mainly Japanese particles (e.g. ga, wa, kara etc.). Subsequently these words were mapped in a training dataset, composed of both why-TS and other-TS. $Tf - idf$ was calculated for each function word and feature vectors were built for each text segment. A classification model was built using LogitBoost and tested on various datasets.

3.3 LogitBoost

LogitBoost is a boosting algorithm with a binomial log-likelihood loss function and is part of the ensemble learning methods. The principle that governs ensemble learning is that combining several models produced by a classification algorithm into an ensemble might guarantee better accuracy than a single classifier, under the condition that the models are different enough to avoid making similar errors. In other words, boosting works by combining weak or base learners into a more accurate ensemble classifier. During the boosting process a number of base classifiers are fitted iteratively to re-weighted data in order to build a strong classifier. With each iteration the weight of the misclassified data points is increased while decreasing that of the correctly classified. Therefore, at each next iteration, the base learner will concentrate on the misclassified samples, working on correctly classifying it. Any algorithm normally used for classification can be employed as the base learner, provided it allows weighting of samples.

In Tanaka's study, decision stumps were used as a base learners since they are was easy to use and gave promising results.

3.4 Adaptation of the Bag of Function Words Method to English

Since the "Bag of function words" method was originaly designed only for Japanese, our previous work was dedicated to implementing this method for English. First and foremost, we had to replace the Japanese part-of-speech tagger with one suited to English. We selected the Standford tagger due to its high accuracy (over 95%). This tagger uses the well known Penn Treebank style containing a total of 36 part-of-speech labels. Following the principle of Tanaka's method, we selected 12 part-of-speech labels that we considered labeled words that fulfilled the three conditions described previously. These parts of speech mainly consist in coordinations, conjunctions, prepositions, modal verbs, pronouns, particles and determiners.

Feature Extraction. The Stanford Tagger [11] is run on all the text segments from the training dataset and the function words are extracted. Afterwards every text segment is mapped in the feature space using $tf - idf$ where the term frequency equals the number of times a function word appears in a text segment, and the document frequency measures in how many different text segments the function word is present. After feature extraction the dataset is thus:

$$\{(\boldsymbol{x_i},\, y_i)\}\,, \quad i = 1, 2, \ldots N \qquad y_i \epsilon\ \{true,\, false\} \tag{1}$$

where $\boldsymbol{x_i}$ is the feature vector for a given text segment i, N is the total number of text segments and y_i indicates if the i-th text segment encodes ($true$) or does not encode causation ($false$).

Experimental Results. The preprocessed training dataset is used to build a classifier by using LogitBoost with decision stumps. The performance of the output classifier was evaluated using 10-fold cross-validation and measuring precision, recall, and F-measure of all the classifiers produced.

Our experiment concluded that the classifier was successful, yielding an average precision of 76.1%, and average recall of 70.6% for text segments encoding causality, respectively 72.6% and 77.9% for text segments that do not encode causality.

Although preliminary results were promising, we think the small datasets used for training and testing might affect the validity of our study. Moreover we want to investigate the potential of other words in the why-classification process.

4 Proposed Method

After further analysis of English syntax we concluded that other parts-of-speech hold precious information for why-type classification: along with the parts-of-speech that we considered as labeling function words, some adverbs also fulfilled the conditions to be considered function words. Adverbs such as “before”, “less” or “only” are frequently present in any kind of text corpora and therefore they are not context-dependent. Moreover, since their number is limited, they successfully satisfy the reasonable feature space condition (section 3.2, 1st condition). Considering the properties of these words, we have decided to add them to our initial feature space. The extraction procedure is detailed in subsection 4.1 . This method will be considered as the first method for our tests.

An analysis on the Second Edition of the Oxford English dictionary [1] shows that, out of the 171476 words, over half of the words are nouns, while about a quarter are adjectives, and about a seventh are verbs. In this respect, we assume that nouns and verbs play an important part when it comes to expressing causality. In contrast, we consider adjectives only bring supplementary descriptive information but do not hold notable causality discrimination properties. Hence including verbs and nouns to our feature space might boost the classifier’s performance providing their number remains limited.

On a first approach we considered including only verbs to our analysis since their number is rather limited. We noticed that for 1000 text segments approximately the same number of distinctive verbs were extracted. Therefore including all verbs will almost triple the dimension of the initial feature space. Moreover, only a small amount of these verbs are eligible candidates for causal expression. Given these results two options presented to us: use a predefined dictionary of causal verbs or attempt to automatically extract significant verbs from the set of verbs present in our training dataset. Although the first option is appealing, it implies using a resource build with the help of a linguist expert. Besides, there exists no record of an exhaustive list of causal verbs, most of them being the fruit of scientific papers that deal with a precise subject [6].

For these reasons, we selected the second option: acquiring causal verbs automatically from corpora. To avoid the dimensionality curse we opted for an a priori feature selection technique. With this technique, we are able to extract verbs that discriminate well between why-TS and other-TS. We believe this list also incorporates a fair amount of causal verbs. A full description of this method can be found in subsection 4.2. Due to the importance of nouns in the English language we decided to implement this method for nouns as well (see subsection 4.2).

4.1 Adverb Extraction and Selection

In order to extract the context-independent adverbs from the corpora, we use WordNet [3] as an external resource that will help identify the eligibility of an adverb. With the help of the Stanford Tagger we gather all adverbs in our corpora and select only those whose root does not correspond to content word. WordNet is only used to verify whether the root is identical to a lemma of a verb, noun or adjective, and exclude the adverb if that is the case. We decide to reject these adverbs because we believe they only have a descriptive role in the sentence, with little or no causality information. Moreover, most of them derive from adjectives (by adding the “-ly” suffix) that we have already excluded from analysis. The entire procedure is easy to implement and fully automated.

The WordNet dictionary is a resource broadly used for research purposes displaying a vast lexical database that can guarantee a good coverage of the English vocabulary. Moreover this dictionary is or will be available for many languages, thus guaranteeing the scalability of the present method.

4.2 Verb and Verb & Noun Extraction and Selection

The extraction process is identical for both verbs and nouns. All existing verbs are selected from corpora and lemmatized using the lemmatizer supplied by MorphAdorner [8]. The initial feature vectors, used only for feature selection purposes, are created by following the same procedure we used in our previous word (see section 3.4) by keeping only verbs. These feature vectors are fed to several a priori feature selection algorithms and the representative lemmas are selected. The lemmas extracted are added to the initial feature space, that contained only function words and selected adverbs. Finally, the final feature vectors, used for classification, are generated with the same method. In this feature vectors all features are represented (function words, adverbs and selected lemmas). We chose to perform the feature selection on lemmas only, because function words and adverbs seem to represent well each text segment due to their redundancy in text. Performing a feature selection on all feature will lead to the elimination of these words and therefore a poorer representation of each text segment.

In our last experiment we follow this procedure for both verbs and nouns. We chose to make the selection on both nouns and verbs at the same time because some of these parts-of-speech share the same lemma (e.g. cause, suggest-suggestion etc.); therefore instead of obtaining two different $tf - idf$ calculations for the same lemma, we obtain only one where the $tf - idf$ value reflects the presence of the lemma in the text and not of the verb or noun individually.

4.3 Feature Selection Algorithms

Feature selection is a data mining technique which consist in choosing representative input features and removing irrelevant and redundant ones. This method is used in supervised learning to find feature subsets that will boost the classification accuracy. Moreover, with fewer features to analyze the classification algorithm will operate faster and more effectively.

For our study we investigated the performance of Correlation based Feature Selection (CFS) and χ^2. The 2 methods differ by the fact that CFS uses one-sided metrics while χ^2 uses two-sided ones. Feature selection algorithms using two-sided metrics select features most indicative of both membership (positive features) and non-membership (negative feature), while feature selection using one-sided metrics only extracts features most indicative of membership.

Correlation Based Feature Selection. CFS uses a heuristic to measure the usefulness of each feature in predicting the class label by considering their average correlation to the class against the average inter-correlation. In other words, a feature has increased importance if it has high average correlation with the class and low inter-correlation with other features. The formula of the heuristic is:

$$G_s = \frac{k\,\overline{r_{ci}}}{\sqrt{k + k\,(k-1)\,\overline{r_{ii}}}} \qquad (2)$$

where k is the number of features in the subset, $\overline{r_{ci}}$ the mean feature correlation with the class, and $\overline{r_{ii}}$ is the average feature-feature inter-correlation.

To determine which features are included in the output subset the heuristics is combined with a search strategy.

χ^2 Based Feature Selection. The χ^2 statistic measures the lack of independence between a word, w, and a given category, c_k. $\chi^2(w, c_k)$ has a natural value of zero if word w and category c_k are independent. Since $\chi^2(w,\ c_k)$ is per-class, the average is used to combine the scores and select the k most representative features.

This method outputs a ranked list of all the variables in the dataset with their respective score. The number of features to include in the final subset is determined empirically.

4.4 Classification Algorithms

To evaluate the performance of the different proposed methods we consider two classification algorithms : LogitBoost and Support Vector Machine (SVM) with a Pearson VII function based kernel (Puk) [12]. LogitBoost has already been used for classification purpose in previous work, a full description being available in section 3.3. Support Vector Machine is a very promising machine learning tool due to its generalization ability and robust behavior over a variety of different learning tasks. However, SVM can perform effectively only if a suitable kernel function is applied. Usually the latter is determined experimentally by applying various kernel functions and selecting the best performing.

In this paper we used Puk function because of its ability to behave as a generic kernel. The Puk function can be varied gradually from a Gaussian bell to a Lorentzian line shape just by changing its input parameters, σ and ω. The Puk kernel function is:

$$K(x_i, x_j) = \frac{1}{\left[1 + \left(\frac{2\sqrt{\|x_i - x_j\|^2}\sqrt{2^{1/\omega} - 1}}{\sigma}\right)^2\right]^{\omega}} \tag{3}$$

In Eq. (4) the parameter σ determines the width (sharpness) of the Pearson VII function. The parameter ω controls the actual shape (tailing) of the function. The Euclidean distance between the two vector arguments is normalized ensuring that all distances between the input objects and the map weights are in the range [0-1]. Due to this uniform rescaling we can easily optimize the kernel function just by modifying the values of σ and ω.

5 Experimental Settings and Results

5.1 Datasets

The data used for the experiment came from three main sources : Yahoo!Answers, Wikipedia and the Why-TS made available by Verbene on her website. From Yahoo!Answers we have randomly extracted text segments that were the answer to a why-question, for the positive data, and also those that were the answer to other types of questions (e.g. when, what, who), for negative data. Only the answers from the best-answer category were selected. From Wikipedia we randomly extracted definitions to serve as negative data in our experiment, and also content-related passages to each why-TS from Verberne's dataset. The latter were extracted manually and served as negative examples that possessed similar word content as the text-segments from the Verberne's dataset.

From the data collection, we constructed the three training datasets displayed in Table 1. For each set the origin of negative/positive data is indicated with the mention whether the data was automatically extracted (A) or manually (M). The data used for training is balanced (same number of why-TS and other-TS). The TS column indicates the total number of text segments used for training.

Table 1. Training datasets

Name	TS	Negative Data	Positive Data
TR-V	432	Verberne Dataset	Wikipedia (M)
TR-Y	2000	Yahoo!Answers (A)	Yahoo!Answers (A)
TR-YW	2000	Yahoo!Answers (A)	Wikipedia (A)

Table 2. Test datasets

Name	Used with	Negative Data	Positive Data
Test-V	TR-V	Yahoo!Answers	Wikipedia
Test-Y	TR-Y	Yahoo!Answers	Yahoo!Answers
Test-YW	TR-YW	Yahoo!Answers	Wikipedia

For testing purposes we constructed incrementally several datasets in order to evaluate the performance of the algorithms with the increase of data. We created test sets of 2000, 4000, 6000, 8000 and 10000 samples. The origin of the data used to test each training dataset is displayed in Table 2. All data was gathered automatically.

5.2 Feature Extraction

The features were extracted from the datasets described in section 5.1. using Stanford Tagger for part-of-speech labeling and MorphAdorner Lemmatizer for extracting the lemma for verbs and nouns. A simple spell corrector algorithm was also used to correct recurrent spelling mistakes. Following the three methods described in section 4. we experimented with six possible feature vectors (see Fig. 1). There are twelve scenarios of the experiments in which three scenarios do not incorporate a feature selection step. The description of each is shown in Table 3.

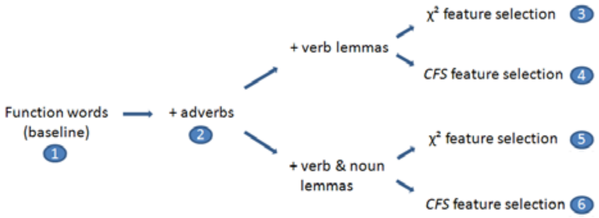

Fig. 1. Possible feature configurations vectors compared in the experiment

5.3 Parameter Optimization

In order to obtain maximum accuracy for the classification models we have to determine the optimal parameter setting for both classifiers. The optimization parameters were: the number of iterations, i, for the LogitBoost algorithm and σ, ω and the complexity parameter, c, for SVM-Puk. We evaluate the parameter setting performance over a 10-fold cross-validation performed on the training datasets; thus, the data used for parameter tunning is independent from the test sets. Table 4 contains the optimal parameter setting we have found.

Table 3. Description of scenarios

Features	Feature Selection	Classifier used	Scenario
Function words (F)	None	SVM - Puk	**F1**
		LogitBoost	**F2**
F + adverbs (FA)	None	SVM - Puk	**FA1**
		LogitBoost	**FA2**
FA + verbs	χ^2	SVM - Puk	**FV1**
		LogitBoost	**FV2**
	CFS	SVM - Puk	**FV3**
		LogitBoost	**FV4**
FA + verbs & nouns	χ^2	SVM - Puk	**FN1**
		LogitBoost	**FN2**
	CFS	SVM - Puk	**FN3**
		LogitBoost	**FN4**

Table 4. Optimal parameter setting

	TR/Test-V				**TR/Test-YW**				**TR/Test-Y**			
Parameters	i	c	ω	σ	i	c	ω	σ	i	c	ω	σ
F1	-	1.4	0.9	1.2	-	1.4	0.9	1.2	-	1.0	1.6	1.6
F2	50	-	-	-	110	-	-	-	200	-	-	-
FA1	-	1.4	2.0	2.3	-	1.4	1.5	1.9	-	0.8	1.1	1.1
FA2	80	-	-	-	110	-	-	-	80	-	-	-
FV1	-	1.4	2.0	2.4	-	1.3	4.0	4.0	-	1.3	2.2	2.8
FV2	90	-	-	-	200	-	-	-	200	-	-	-
FV3	-	1.4	2.5	2.5	-	1.0	1.6	2.0	-	1.1	0.9	1.1
FV4	100	-	-	-	200	-	-	-	200	-	-	-
FN1	-	1.2	3.0	3.0	-	1.2	2.0	2.2	-	1.2	2.0	2.1
FN2	100	-	-	-	200	-	-	-	300	-	-	-
FN3	-	1.5	4.0	4.0	-	1.1	2.5	2.5	-	1.4	1.6	1.5
FN4	95	-	-	-	200	-	-	-	300	-	-	-

5.4 Results

All twelve scenarios were executed on each of the three training databases. To estimate the performance of the model built with each scenario we use a 10-fold

Table 5. Results obtained using the SVM classifier. Percent improvement, as well as statistical significance is with respect to the SVM baseline (F1).

Scenario	**TR/Test-YW**	**TR/Test-V**	**TR/Test-Y**
F1 (baseline)	0.9108	0.8101	0.6418
FA1	0.9178 (0.70%)	**0.8318 (2.17%)**	0.6467 (0.49%)
FV1	0.9126 (0.18%)	0.8196 (0.95%)	0.6602 (1.84 %)
FV3	0.9158 (0.50%)	*0.8082 (-0.19%)* †	0.6514 (0.96%)
FN1	**0.9252 (1.44%)**	*0.7700 (-4.01%)*	**0.6654 (2.36%)**
FN3	0.9198 (0.90%)	*0.7992(-1.09%)*	**0.6654 (2.36%)**

Table 6. Results obtained using the LogitBoost classifier. Percent improvement, as well as statistical significance is with respect to the LogitBoost baseline (F2).

Scenario	**TR/Test-YW**	**TR/Test-V**	**TR/Test-Y**
F2 (baseline)	0.9356	0.5344	0.6326
FA2	0.9381 (0.25%)†	0.6490 (11.46%)	0.6410 (0.84%)
FV2	0.9432 (0.76%)	**0.6722 (13.78%)**	0.6432 (1.06%)
FV4	0.9432 (0.76%)	0.5758 (4.14%)	0.6440 (1.14%)
FN2	**0.9496 (1.40%)**	0.6300 (9.56%)	**0.6556 (2.30%)**
FN4	0.9428 (0.72%)	0.6434 (10.9%)	**0.6556 (2.30%)**

cross-validation. Once each model has been optimized over cross-validation, we perform the evaluation tests on test datasets.

Tables 5 and 6 contain the results of our findings. The displayed value represents an average of the 5 F-measures (for 2000, 4000, 6000, 8000 and 10000 text segments) we measured for each scenario during our experiment. A significance paired t-test was performed on the 5 F-measure scores measured for each scenario, and succeeded on almost all at a $p < 0.05$ level; the scenarios that passed the test only at the $p < 0.1$ level are denoted with a †. In order to determine the most significant features for the CFS method we used a hill climbing search algorithm; for the χ^2 selection process we selected all features that had a score superior to zero.

Results show that all 3 methods over-perform baseline, with one slight exception for the $TR/Test-V$ with SVM classifier group (refer to the results in italic from table 5). In this case both function words and function words plus adverbs yield better results than the methods that integrate verbs or verbs & nouns. We believe this is a consequence of the fact that negative data was built with similar content words that existed in the positive data. Therefore verbs and nouns have lost their discriminative power when integrated in the SVM classification model. On the contrary, the LogitBoost models built for this set (FV2, FV4, FN2, FN4) are less affected by the content similarity and perform better than baseline.

Both LogitBoost and SVM are successful classification models on all data, yielding similar performance, except for the $TR/Test - V$ data where SVM classification outperforms LogitBoost with over 20% (refer to second column of tables 5 and 6). In terms of feature selection χ^2 and CFS give similar results. While χ^2 is faster in ranking the results, CFS is easier to manipulate since we are not required to determine the cut-off value that would produce the best results. Globally we notice the verbs & nouns methods (FN) are the best performing ones except for the TR-V Test-V data. Results show that all methods discriminate very well between random definitions and why-TS (up to 94%) while applied to a more heterogeneous database the accuracy of classification falls down to 65%.

In terms of execution time we notice that the average speed decreases with the number of features that are included in the analysis, but also with the number of validation and training instances. Therefore the time to build the model varies from 2.7 seconds, on TR-V, to 115.5 seconds, on TR-Y, for the LogitBoost classifier and from 190 milliseconds, on TR-V, to 28.5 seconds, on TR-Y, for the SVM classifier. The worst execution time with respect to testing is obtained when performing the test on the 10000 instances on the Test-Y dataset; the time is of 1.97 seconds with a LogitBoost classification model and of 70.38 seconds with a SVM-Puk classification model.

We show the progression of our two classification models with the increase of test data in Fig. 2 for SVM-Puk and respectively Fig. 3 for LogitBoost; the dataset used in both figures is TR/Test-Y. We have excluded the FV1-2 and FN1-2 because we stated before that the χ^2 feature selection performance is similar to CFS while CFS is easier to manipulate.

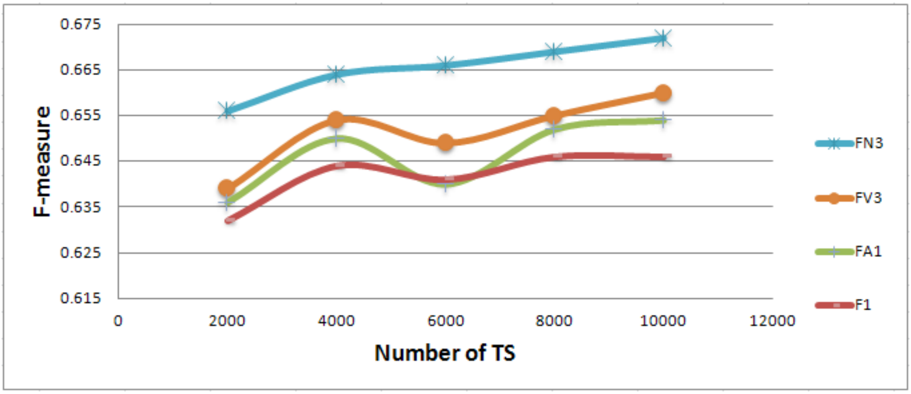

Fig. 2. F-measure value at various test dataset sizes for SVM-Puk Classifier

This graphics prove that the FN scenario is the best performing with both SVM and LogitBoost. We note that the SVM-Puk classfier is very sensitive

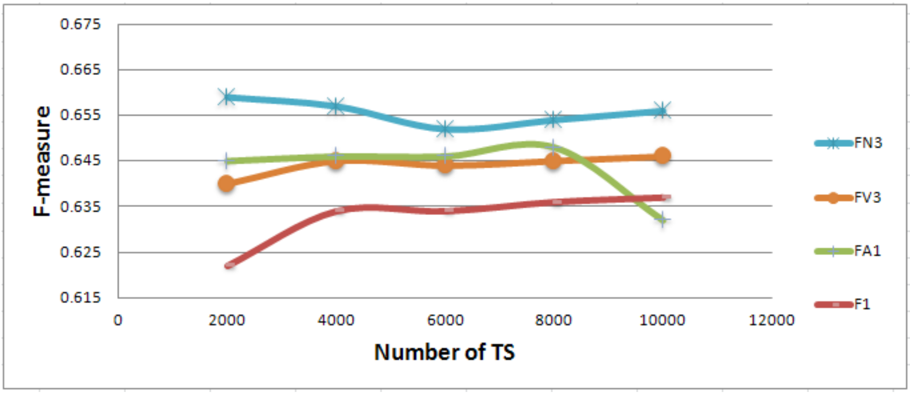

Fig. 3. F-measure value at various test dataset sizes for LogitBoost Classifier

to the quality of the test data, while the LogitBoost classifier suffers very little from it.

5.5 Conclusion and Future Works

In this paper we investigated several methods to improve the performance of the “Bag of function words” on English. Through our work we have shown the importance of adding new features (adverbs, verb lemmas and verb & noun lemmas) in boosting the classification of why-text segments. Initially, context-independent adverbs were added to the features showing small but valuable improvement of classification accuracy on all test datasets. Taking into account the amount of nouns and verbs in the English language we assumed they held significant information in terms of expressing causality and hence considered integrating them in the analysis. Confronted with their large number, we have added a feature selection step to our method to avoid the dimensionality curse. Adding the features selected by the feature selection algorithm has proven successful improving the classification performance with approximatively 2.5% for nouns & verb lemmas and 1% for verb lemmas.

We are tempted to think SVM with a Puk kernel might be a more appropriate classifier than LogitBoost since it can be parameterized to adapt to any kind of data and the results show that SVM slightly outperforms LogitBoost for most of the validation tests performed, but we believe this matter requires further investigation. However, during these experiments the optimum configuration parameters were determined by a local search performed manually. Therefore the accuracy of this classification model can be further improved by applying an automatic extensive search for the configuration parameters.

Future work will be dedicated to making our approach more robust to answers that contain noise (spelling mistakes, emoticons) and also handling answers that do not contain direct answers but an url to further resources.

References

1. AskOxford. How many words are there in the english language?, `http://www.askoxford.com`
2. Blanco, N., Castell, E., Moldovan, D.: Causal relation extraction. In: Proceedings of the Sixth International Language Resources and Evaluation, LREC 2008 (2008)
3. Fellbaum, C.: WordNet: An Electronic Lexical Database. Bradford Books (1998)
4. Girju, R.: Automatic detection of causal relations for question answering. In: Proceedings of the ACL 2003 workshop on Multilingual summarization and question answering, pp. 76–83 (2003)
5. Higashinaka, R., Isozaki, H.: Automatically acquiring causal expression patterns from relation-annotated corpora to improve question answering for why-questions. ACM Transactions on Asian Language Information Processing (TALIP) 7(2), 1–29 (2008)
6. Khoo, C., Chan, S., Niu, Y.: Extracting causal knowledge from a medical database using graphical patterns. In: In Proceedings of 38th Annual Meeting of the ACL, Hong Kong, pp. 336–343 (2000)
7. Nagy, I., Tanaka, K., Takiguchi, T., Ariki, Y.: Extracting why text segment from web based on grammar-gram. In: Proceedings of the Fouth Spoken Document Processing Workshop (2010)
8. Philip, R.: "Pib" Burns of Academic and Northwestern University Research Technologies. English lemmatizer, `http://morphadorner.northwestern.edu/morphadorner/lemmatizer/`
9. Tanaka, T., Takiguchi, K., Ariki, Y.: Automatic why text segment classification and answer extraction by machine learning (japanese). Journal of Information Processing Society 49(6), 2234–2242 (2008)
10. Tanaka, T., Takiguchi, K., Ariki, Y.: Domain independent why text segment classification and answer extraction by grammar-gram and grammarverb-gram (japanese). WI2, pages pp. 89–94 (2009)
11. Toutanova, K., Christopher, D.: Manning. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In: Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora, pp. 63–70 (2000)
12. Ustun, W.J., Melssen, B., Buydens, L.M.C.: Facilitating the application of support vector regression by using a universal pearson vii function based kernel. Chemometrics and Intelligent Laboratory Systems 81, 29–40 (2006)
13. Verberne, S.: Developing an approach for why-question answering. In: EACL 2006: Proceedings of the Eleventh Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop, pp. 39–46 (2006)
14. Verberne, S., Boves, L., Oostdijk, N., Coppen, P.-A.: Evaluating discourse-based answer extraction for why-question answering. In: SIGIR 2007: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 735–736 (2007)
15. Verberne, S., Boves, L., Oostdijk, N., Coppen, P.-A.: Using syntactic information for improving why-question answering. In: COLING 2008: Proceedings of the 22nd International Conference on Computational Linguistics, pp. 953–960 (2008)
16. Verberne, S., Boves, L., Oostdijk, N., Coppen, P.-A.: What is not in the bag of words for why-qa? Comput. Linguist. 36(2), 229–245 (2010)