Regular Paper

Extracting Why Text Segment from Web Based on Grammar-gram

Iulia NAGY ,[†1,†2] Katsuyuki TANAKA,[†1]
Tetsuya TAKIGUCHI [†1] and Yasuo ARIKI[†1]

In the current project, we aim at developing a novel approach for automatically answering why-questions in English. In order to achieve our goal, we explore one of the methods described in the literature: Grammar-gram and grammarverb-gram why extraction procedure using domain-independent text answer segment. The existing research only addresses Japanese data, therefore our intention is to adapt and improve it so that it could be suitable to English. Taking into account the fact that there are very significant differences between English and Japanese, particularly in sentence and grammar structure, our current attempt consists in analyzing the effectiveness of this method on English.

1. Introduction

As nowadays Internet represents a major source of information, many people rely on it in order to acquire the knowledge they are looking for. Nevertheless, obtaining the right information can turn into a tedious task that consists in consulting, with or without success, various web pages presented by a search engine. In order to facilitate the process of finding information over the Internet, the idea of creating a question answering service, which provides precise answers to specific questions, emerged. Even though existing research has already produced a considerable number of satisfying QA systems for factoid questions, the progress in the domain of non-factoid question remains rather limited. Therefore our attention focuses on creating a QA system for non factoid questions, more precisely a why-QA system.

†1 Kobe University
†2 INSA de Lyon, France

The most common approach undertaken by researchers when it comes to building a why-QA system is to use hand-crafted patterns to extract viable passages that represent an answer to a why-question. Although this method has proven its effectiveness on English, it still remains labor intensive and domain-dependent, since rules are mostly hand-coded. Moreover, due to the various ways used to express cause, it is almost impossible for hand-crafted methods to insure a full coverage of these causal expressions.

The explosion of community portals such as Yahoo!Answers and WikiAnswers allowing users to post questions and/or answer questions asked by other members of the community, has been of great aid to QA system development. The large number of question-answer pairs hosted on these portals enabled automatic training methods to emerge. These methods were formerly inapplicable due to the lack of data.

Unlike rule-based methods, machine learning methods have the advantage of acquiring knowledge in a less time and effort consuming manner. This is due to the fact that rule-based methods require manual extraction and validation of efficient rules, while machine learning methods only need to automatically extract causal expressions from corpora in order to derive causal expression patterns.

The purpose of our research is to build an efficient why-QA system able to detect why text segments from arbitrarily built corpora. A text segment is a group of sentences that are an eligible candidate for answering a why-question. Since the classifier is a fundamental component of any automatic QA-system, we believe building it automatically is a crucial step to improve the performance of these systems. Therefore, our attention has focused on building automated methods to build classifiers. Scientific literature presents numerous such methods, but we were particularly interested in an approach described in the Japanese literature[6]. This method is based on a bag of grammar approach, and uses machine learning to build fully automated classifiers. In the present paper, we present the changes made to the method we have chosen, evaluate and discuss its effectiveness on English.

This article is organized as follows: Section 2 describes the related work on why-QA, Section 3 describes the method we inspired our work from. Section 4 presents the adaptation process that the base method has undergone, while

Section 5 describes the experimental preparation and the results. Finally Section 6 contains the conclusion and the description of future works.

2. Related Work

Two main trends in creating the mechanism for why-QA system exist in the current research: the Rule Based method, using predefined rules, and the Machine Learning approach, which proposes automatic rule extraction.

The Rule Based method consists in creating a set of rules out of patterns that have been detected in the analyzed corpus. This corpus ought to contain large amounts of features that express cause for the rules extracted to be correct. This method is one of the first used for creating why-QA systems[8], and has undergone significant progress in the past years.

One of the best known figures applying a Rule Based technique in the domain of why-QA is Verberne[10)–12)]. Her initial work was based on retrieving why-answers by making use of the Rhetorical Structure Theory. Recently she proposed a re-ranking method for paragraphs retrieved by Wumpus, in which the initial results are ranked by QAP algorithm. Verbene argues that the frequency variables used in QAP ranking do not reflect the importance of each term in the examined paragraph. She believes that information extracted from the syntax of the analyzed QA pair is crucial to improving ranking. Thus, she proposes, as a re-ranking method, to weight the score assigned to a QA-pair by QAP with a number of syntactic features. These features, 31 in total, mainly include information about the syntax and semantics of the analyzed phrase, synonyms and a list of words expressing cause. Since the syntactic structure is needed for the majority of features, two type of parsers are used to extract it : the Pelican (constituency parser) and the EP4IR parser (statistical parser). Values that have been manually extracted from each parser's output are assigned to the 31 features examined. These values reflect the importance of one feature in the QA-pair. Verbene also manually selected the correct parser tree (gold standard) out of the various parser trees produced by Pelican. In order to determine the optimal weighting scheme of the features for improving rank, Genetic algorithm was used. The initial weight of each feature is trained by using the values that were assigned to it by the gold standard parse tree. By integrating these values into the ranking algorithm, the result has significantly improved compared with baseline.

Although it provides an effective re-ranking for why-answers, this method requires a deep syntactical and semantical analysis of the language, implying a very solid knowledge of grammar and linguistic. Since the tools used in the process are essentially only adapted to English, and the process itself requires advanced language processing skills (e.g. determine the choice of features), this method remains hardly adaptable to a large range of languages.

While being more robust and less labor-intensive, most of the Machine Learning approaches tend to have limitations in that they only provide answers containing casual verbs[3)] or containing a specific list of relators[1)]. A new, more general, method has been introduced by Higashinaka and Isozaki[4)].Their approach consists in acquiring causal expression patterns automatically, by making use of the Japanese EDR dictionary. This resource contains phrases gathered from heterogeneous sources that have been manually labeled with their semantic role. The information thus obtained provides a relation annotation, indicating the type of the terms in the phrase (e.g terms indicating cause, indicating object etc.).

Higashinaka and Isozaki extracted from the dictionary's corpus all structure that was annotated as a causal relation and replaced the terms that did not indicate a cause and that were context dependent (e.g. nouns, verbs, adjectives etc.) by a "*". The structure left mainly contained Japanese function words. Subsequently, all the clause structures captured in that manner, along with features designed from manually extracted rules used to point the cause, served to train a ranker by machine learning.

While their system is said to be the best-performing fully implemented why-QA system for Japanese, it has some points that can be subject to discussion. The entire system is based on the knowledge provided by the EDR dictionary. Consequently, the system is not fully automated since it extracts the information from a hand-crafted resource. Furthermore, the EDR might not available in wide range of languages and is rather high-priced.

Other research papers that address the subject of why-QA systems are those of Tanaka[5),6)]. His first approach[5)]consisted in training a classifier by machine learning out of a bag-of-words (BOW) feature space. Since all the words in the answers are represented in the feature space the dimension of the feature vectors

is rather imposing and therefore the classifier might be subjected to noise. Also, the data used for the training is domain oriented since it is hardly possible to capture the diversity of a language in a reasonable number of text segments (most of the data contains nous and verbs that reflect specific information). Despite its limitations, this method's interest resides on the fact that it can be implemented for any language since no previous syntactical or semantical knowledge of the data is required.

In his following attempt[6] , Tanaka casted aside the domain dependent terms (adjectives, noun and verbs) and focused uniquely on the function words. This method, called hereafter "Bag of function words" method, solves the domain-dependency problem of the BOG approach. This fully automated methodology enables to build domain-independent classifiers by using data obtained from the Internet.

After comparing several machine learning algorithms we decided to investigate more carefully the "Bag of function words" method. The basis of this method is quite simple: by extracting function words out of a text corpora it is possible to build a classifier that can detect why text segments on any type of input data. Moreover, the performance of the method in terms of effectiveness and accuracy of classification is convincing. Thus, we chose to adapt this approach for English and test its effectiveness.

3. Bag of function words method

3.1 Preliminary remarks

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.

3.2 Method outline

"Bag of function words" proposes a machine learning approach for automatically building domain independent classifiers for why-text segment. Tanaka considers that the following 3 conditions need to be fulfilled when building a domain independent 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.

Tanaka pointed out that only a specific class of words satisfies these 3 conditions: the function words class. These words are usually put aside when creating a word retrieval system because of their abundance in the text and their lack of discriminative power when it comes to the context. Yet, they hold very precious information when it comes to identifying the type of a phrase (definition, cause, explanation etc.) and in consequence satisfy the 3rd condition. Also, when extracted out of their initial context they lose all meaning which makes them eligible for the generality requirement (2nd condition). Moreover, since their number is limited they also fulfill the first condition.

In the learning process, training data (containing correctly labeled encoding or not encoding causation data) is mapped to the feature space. Once the feature vectors have been created, and labels were assigned to each data record, the classifier construction problem is solved by applying a supervised learning algorithm, in this case LogitBoost. This algorithm will be discussed in detail in subsection 3.3. The weak learners provided by LogitBoost will learn effective features, that discriminate well between WTS and NWTS, and produce a classifier.

3.3 LogitBoost

LogitBoost learning algorithm is a statistically-based boosting procedure that makes it possible to train accurate classifiers. Boosting procedures have recently become popular because they are simple, elegant, powerful and easy to implement.

Boosting is a classification scheme that works by combining weak learners into a more accurate ensemble classifier.

More precisely it combines a large number of weak learners through the medium of a weighted sum.

A classification procedure is iteratively applied to the weighted feature vectors in the dataset. Initially each feature vector is assigned an equal weight. At each iteration, the learner tries to build a weak classifier according to the performance of the previous weak classifiers. During the learning process, the weights of the feature vectors that are classified incorrectly are increased, while those belonging

to correctly classified feature vectors are decreased. The purpose of this approach is to give increased importance to those vectors on which the previous classifier fails by re-feeding them to the weak learner that performs accurately on examples that were hard for the previous weak learners.

One of the best known boosting procedures is AdaBoost. Although AdaBoost had very good generalization (the ability to classify new data), it suffered from the over-fit problem when dealing with very noisy data. In order to correct this deficiency, Friedman et al.[2], proposed LogitBoost.

LogitBoost can be pictured as an additive logistic model (equation 1) that uses Newton step for minimizing the exponential criterion (equation 2). LogitBoost supports multi-class classification by producing a classifier in each class. LogitBoost is proposed as optimizing the log-likelihood for fitting logistic model. In each class, posterior distribution is calculated and weight of weak learns are re-computed based on the distributions.

As a result, it produces functions in J classes $\{F(x)_j \,|\, j = 1 \ldots J\}$and output the classifier by $argmax_j \, F_j \, (x)$.

$$F \, (x) = \sum_{m=1}^{M} c_m \, f_m \, (x) \tag{1}$$

where c_m are the constants to be determined and f_m are basis functions

$$J \, (F) = E \left(e^{-yF \, (x)} \right) \tag{2}$$

Besides providing high learning speed and performance, the major benefit of using LogitBoost consist in its adaptability to a multi-class classification, allowing an evolution of the system toward a general non-factoid QA system.

4. Adaptation of the presented method

Following the methodology introduced in the previous section we have implemented a similar system for English.

4.1 Building training data

(1) **Data**

The data needed to train the classifier is extracted from text segments that have been acknowledged as why-answers, for the positive examples, but also text segments that do not encode causation, for the negative examples.

(2) **Pre-processing**

In order to identify function and content words, all the text segments need to be annotated with part of speech (POS) tags. From the list of POS proposed by the POS Tagger algorithm used, a list of functional and content related POS is established.

Since the POS Tagger for Japanese is not adapted to English, an appropriate software for English was necessary. After a short analysis of various POS Tagger proposed for English, on criterion such as performance, simplicity, speed, we leaned toward the Stanford tagger[9]. Offering extended precision in labeling (over 95%), the Stanford tagger embodies only 36 tags whose complete description can be found in Santorini's work[7].

Once it has been decided which are the functional parts of speech, only words whose tag falls into this category are extract from the training data.

(3) **Feature Extraction**

Since the main purpose of this method is to create a classifier using the supervised machine learning approach, the feature vectors have to be defined.

The choice of the appropriate algorithm for the data mining step is dominated by the aspect of feature set. In this case, the feature set contains all the function words that were extracted from training data. The dataset $\{TS^t \,|\, x^t, \, r^t\}^{t=1 \ldots n}$ for training, learning and testing data is composed of causation encoding text segments, and non-encoding causation text segments (defined as WTS and NWTS in Tanaka's paper[6]), along with the relation they represent (causation, non-causation) represented by r in the definition.

After performing feature extraction on the given dataset, every text segment record is denoted by a numerical feature vector and a class label is assigned to the record.

$$\{(\vec{x_i}, \, y_i)\}, \quad i = 1, 2, \ldots N \qquad y_i \epsilon \, \{true, \, false\} \tag{3}$$

where $\vec{x_i}$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-est text segment encodes ($true$) or does not encode causation ($false$).

Each text segment from the dataset is mapped into the feature space by using the term frequency and inverse document frequency ($tf - idf$) for each function

word present in the analyzed text segment. The algorithm used for calculating $tf - idf$ will be explained in subsection 3.2.

(4) **Using tf-idf**

$Tf - idf$ weight is often used in information retrieval and text mining, since it allows a good representation of the importance of the analyzed words. In this method, this calculation determines how relevant a given function word is in text segment. Function words that are common in a small group of text segments tend to have higher $tf - idf$ value than common function words such as articles. Therefore, function words that encode causality will see their importance increased and serve to produce an accurate why-classifier.

Given a text segment collection TS, a function word t_i, that is represented in the feature set, and an individual text segment ts_j, we calculate

$$(tf - idf)_{i,j} = tf_{i,j} \times log\frac{|TS|}{df_i} \quad (4)$$

where $tf_{i,j}$ (term frequency) equals the number of times t_i appears in ts_j, and df_i (document frequency) equals the number of text segments in which t_i appears. The base of the logarithm that has been used is base 10. The value obtained for each function word analyzed, $(tf - idf)_{i,j}$, is between 0 and 1, and will be mapped in the feature vector of the corresponding text segment.

4.2 Adjustments necessary for English

As stated is the previous section, Tanaka defined function and content words, that are more commonly delimited in English linguistics as two classes : open and closed words classes. To get a better understanding of what these terms refer to, we have listed them in **Table 1**.

Unlike Japanese, English frequently does not mark words as belonging to one part of speech or another. Words like fly, break, cause might all be either verb forms or nouns. Even though "-ly" usually indicates the presence of an adverb, not all adverbs end in "-ly" and not all words ending in "-ly" are adverbs.

Also major differences in the phrase construction can be seen : English forms phrases by adding new words one at a time at the beginning of previously-constructed phrases. By contrast, Japanese forms phrases by adding new words at the end.

Since Japanese differs significantly from English, and English tends to be more ambiguous than Japanese when it comes to identifying the part of speech corresponding to a word, increased attention has been given to the selection of the parts of speech that define a function word. **Table 2** contains the classification of parts of speech into function (BOG) and content (BOW) part of speech that has been used in the experience. In order to distinguish correctly function word and content words in English we analyzed the description given to each part of speech in Santorini's work[7]. Only the words that fullfield the 3 conditions described in subsection 3.2 were labeled as function words, and exploited later in the analysis. This selection process might not be optimal, and will be the subject of a thorough analysis in future works. Due to issues of brevity, only the POS tags are listed in the table, their full description being available in Santorini's work.

Table 1 Word classes

Open word class	Closed word class
adjectives, adverbs, interjections, nouns, verbs (except auxiliary verbs)	auxiliary verbs, clitics, coverbs, conjunctions, determiners (articles, quantifiers, demonstrative adjectives,possessive adjectives), particles, measure words, adpositions (prepositions, postpositions, circumpositions), preverbs, pronouns, contractions, cardinal numbers

5. Experiments and Evaluation

5.1 Data

As data for developing and testing our system for why-QA, we had 2 major sources :

- 400 randomly selected why-questions from the Webclopedia set (questions asked to the online QA system answers.com, gathered by Hovy et al.) and for each question a Wikipedia text fragment giving the answer and a pointer

Table 2 Declination of POS used for English

Feature	Example	POS
BOG	for, because, the, which, to	CC, DT, EX, IN,MD, PDT, RP, TO, WDT, WP, WP$, WRB
BOW	all morphemes	all other 24 POS described in the cited work

to the complete Wikipedia document, that was made available by Verbene on her website; Out of this data, only 216 entries were used, since all the others did not have a valid answer. Those entries represent the positive examples, encoding causality, of our dataset.

- in order to obtain our negative data, where answers do not encode causality, we extracted 216 valid definition phrases from Wikipedia. These definitions were randomly selected since we are only interested in the function words that they contain.

The extracted text segments were stored in files that indicated for each text segment the words it contained with their respective POS labels, and also the value of the text segment, true or false, if it encoded or not causality.

5.2 Extracting features

Each sentence in the 432 text segments corpora was parsed using Stanford POS Tagger. Only the feature whose part of speech tag was included in the BOG feature described in Table 2 have been included in the feature set. We have so obtained a feature set that contains 121 features. In addition to extracting the features, we counted the term frequency and document frequency of each term encountered.

Once the feature set and the corresponding term and document frequency for each term were collected, we re-iterated through our dataset and mapped each text segment into a feature space by forming a feature vector with $tf - idf$ as elements. The $tf - idf$ was calculated by using the algorithm indicated in subsection 3.4.

5.3 Experiment

Following the methodology described in section 3, the classifier is constructed by applying LogitBoost using decision stumps on the dataset. The LogitBoost algorithm that was used in this experiment, is provided in the data mining software Weka[13]. Decision stumps split on only one attribute, so they are robust against over-fitting, and can be understood easily. We used the default parameters of LogitBoost and only modified the number of iterations. The procedure for building the classifier was iteratively applied 5 times, each time increasing the number of iterations by 50. Our starting point was 50 iterations. Therefore we produced 5 why-classifiers on 50, 100, 150, 200 and 250 iterations for each dataset. To evaluate the performance of the predictive model we used 10-fold cross-validation and measured precision, recall, and F-measure of all the classifiers produced. Models were trained on nine folds and tested on one. In the present experiment precision corresponds to the number of text segments properly classified over the total number of text segments. Recall corresponds to the number of text segments properly classified over the total number of text segments expected. The results of the experiment are shown in **Table 3** and they globally indicate that the method is effective on English. We have obtained the best precision and recall for 50 iterations, but since the size of the dataset is rather small and the difference compared to the other results is minimal, we cannot state with certainty that the classifier for 50 iteration is optimal in this case. In order to evaluate which classifier is optimal, we believe the experiment should be repeated on a significantly larger dataset.

Table 3 Detailed results of the experiment

No of iterations	Type of TS	Precision	Recall	F-measure
50	WTS	0.776	0.736	0.755
	NWTS	0.749	0.787	0.767
100	WTS	0.759	0.699	0.728
	NWTS	0.721	0.778	0.748
150	WTS	0.769	0.708	0.737
	NWTS	0.730	0.787	0.757
200	WTS	0.764	0.704	0.733
	NWTS	0.725	0.782	0.753
250	WTS	0.739	0.681	0.708
	NWTS	0.704	0.759	0.719

5.4 Evaluation

Table 4 resumes the findings of our experiment, by showing the average results obtained for our three parameters : precision, recall and F-measure.

The system correctly classified 321 instances out of 432, 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.

Even though only a fairly small training corpus of 432 examples were used for

this experiment, the method performs reasonably well. We conclude that the method proposed in section 3 is effective on English.

Although we believe the performance of the classifier will improve if the size of the training corpus is increased, this investigation is left as a future work.

6. Conclusion

The approach presented in this paper for the creation of automatical QA-system is an application to English of an already existing methodology for Japanese. The salience of this method resides in the fact that it provides a domain independent and easy to implement classifier for why-QA systems.

Our experiment has proven that a very similar implementation of the previously cited method yields convincing results on English.

A key element to really investigate the potential of the method would be to apply it on a substantial dataset, containing at least 5000 examples.

As a future work, we intend to exploit the causative constructions in English, that may contain verbs and nouns, by a similar automatic extraction from corpora.

References

1) Nuria Castell Eduardo Blanco and Dan Moldovan. Causal relation extraction. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*. European Language Resources Association (ELRA), May 2008. http://www.lrec-conf.org/proceedings/lrec2008/.

2) Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Additive logistic regression : A statistical view of boosting. *Annals of statistics*, 28(2):337–407, 2000.

3) Roxana Girju. Automatic detection of causal relations for question answering. In *Proceedings of the ACL 2003 workshop on Multilingual summarization and question answering*, pages 76–83, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

4) Higashinaka Ryuichiro and Isozaki Hideki. Automatically Acquiring Causal Expression Patterns from Relation-annotated Corpora to Improve Question Answering for why-Questions. *ACM Transactions on Asian Language Information Processing*, 7(2):1–29, 2008.

5) Yasuo Ariki, Katsuyuki Tanaka, Tetsuya Takiguchi. Automatic why text segment classification and answer extraction by machine learning (Japanese). *Journal of Information Processing Society iIPSJj*, 49(6):2234–2242, 2008.

6) Yasuo Ariki, Katsuyuki Tanaka, Tetsuya Takiguchi. Grammar-gram and grammarverb-gram why extraction procedure using domain-independent text answer segment (Japanese). *WI2*, pages pp89–94, 2009.

7) Beatrice Santorini. Part-of-speech tagging guidelines for the (penn treebank project), 1990. (3rd revision, 2nd printing).

8) Rohini Srihari and Wei Li. Information extraction supported question answering. In *In Proceedings of the Eighth Text REtrieval Conference (TREC-8)*, pages 185–196, 1999.

9) Kristina Toutanova and ChristopherD. 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*, pages 63–70, Morristown, NJ, USA, 2000. Association for Computational Linguistics.

10) Suzan Verberne. Developing an approach for why-question answering. In *EACL '06: Proceedings of the Eleventh Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 39–46, Morristown, NJ, USA, 2006. Association for Computational Linguistics.

11) Suzan Verberne, Lou Boves, Nelleke Oostdijk, and Peter-Arno Coppen. Evaluating discourse-based answer extraction for why-question answering. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 735–736, New York, NY, USA, 2007. ACM.

12) Suzan Verberne, Lou Boves, Nelleke Oostdijk, and Peter-Arno Coppen. Using syntactic information for improving why-question answering. In *COLING '08: Proceedings of the 22nd International Conference on Computational Linguistics*, pages 953–960, Morristown, NJ, USA, 2008. Association for Computational Linguistics.

13) Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Francisco, CA, 2. edition, 2005.

Table 4 Global results

	Precision	Recall	F-measure
WTS	0.761	0.706	0.732
NWTS	0.726	0.779	0.749