Inter-Sentence Features and Thresholded Minimum Error Rate Training: NAIST at CLEF 2013 QA4MRE

Philip Arthur, Graham Neubig, Sakriani Sakti,
Tomoki Toda, and Satoshi Nakamura

Nara Institute of Science and Technology,
8916-5, Takayama-cho, Ikoma-shi, Nara 630-0192 JAPAN
{philip-a,neubig,ssakti,tomoki,s-nakamura}@is.naist.jp

Abstract. This paper describes the Nara Institute of Science and Technology's system for the main task of CLEF 2013 QA4MRE. The core of the system is a log linear scoring model that couples both intra and inter-sentence features. Each of the features receives an input of a candidate answer, question, and document, and uses these to assign a score according to some criterion. We use minimum error rate training (MERT) to train the weights of the model and also propose a novel method for MERT with the addition of a threshold that defines the certainty with which we must answer questions. The system received a score of 28% c@1 on main questions and 33% c@1 when considering auxiliary questions on the CLEF 2013 evaluation.

Keywords: discriminative learning, minimum error rate training, linear feature model, question answering, machine reading, inter-sentence features

1 Introduction

While years of research on Question Answering (QA) have greatly improved the state-of-the-art, we know that this problem is far from solved. Question answering campaigns such as CLEF [9] and TREC [3] have resulted in a large number of distinct proposals about how to build robust systems that can provide correct answers in the general domain. One of the features of QA that is widely accepted is that "two heads are better than one" [2]. By combining different information sources, we gain the ability to cover up the disadvantages of one system with another information source, which results in more effective QA on the whole. One way to combine multiple systems is to weight each system's score with some value and choose the maximum value from a linear combination [12]. Another important aspect of QA is that it is sometimes good not to answer the question [8]. Many systems currently return No Answer (NoA) if they are not confident because a wrong answer is often worse than no answer [1]. Our system for the CLEF QA4MRE this year is based on these two principles, devising a

number of features that provide useful information to identify the correct answer, and combining them together with a learning framework that is also able to learn when not to answer questions.

We introduce several new features that span multiple sentences in addition to more traditional features such as cosine similarity. These features are combined in a framework that learns both *how* and *when* to answer questions in a single weighted linear model. In particular, we find *how* to answer questions by learning appropriate weigths for each feature, with final score of an answer being their weighted linear combination. We define *when* not to answer by not returning candidates for which scores are less than a set threshold t from other candidates. Finally, we propose a method to intelligently weight the features and threshold using minimum error rate training.

As results for the 2013 evaluation, our best run in main task scored 28% for only main questions and 33% when auxiliary questions are also included.

2 System Description

2.1 Architecture

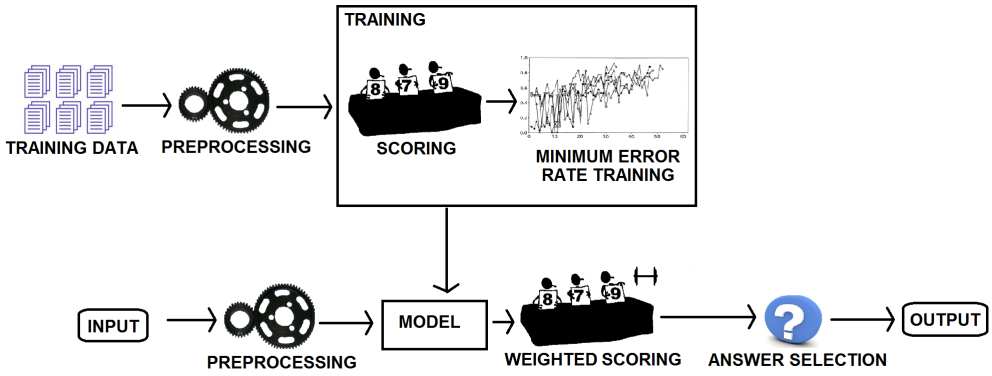

Fig. 1. System Architecture

Our QA system has a modular pipeline architecture, making it easy to modify some modules without changing other parts of the system. The system is divided into three major modules: preprocessing, scoring, and answer selection.

2.2 Preprocessing

As the raw input is full of noise that can drastically reduce the ability to generate appropriate hypotheses, preprocessing is necessary to process the text into

machine-readable format. First, the *tokenization* step splits the sentence into a list of tokens. Second, the *named entity recognition* step uses the Stanford named entity annotator[1] to recognize the existence of named entities. We use the Stanford 4 tag set, which consists of PERSON, LOCATION, ORGANIZATION, and MISC. Third, the *anaphora resolution* module is based on the "last introduced named entity" constraint [5]. Next, the *lowercasing* step alters all tokens into lowercase form and the *stop word deletion* step deletes all words that appear in a stop word list.[2] Finally, *stemming* uses the Porter stemmer to reduce all conjugated forms of some words into their stem.

In addition, we made a few refinements to choose which named entity best replaces the Referring Expression (RE). This module focuses on RE that represent humans according to the following rules:

- "They" and "we" are replaced by the most recent ORGANIZATION named entity.
- "I" is replaced by the first PERSON entity that occurs in the passage, which is generally the speaker.
- "You" is replaced "theaudience."
- "He" and "she" are replaced by the most recent PERSON named entity.
- "It" may refer to either LOCATION, ORGANIZATION, or MISC.
- The remaining pronouns are simply left unaltered.

2.3 Scoring

The scoring criterion is based on linear combination of several weighted features, [12]

$$\mathrm{s}(a_{j,k} \mid q_j, D) = \sum_{i=1}^{n} w_i * \phi_i(a_{j,k}, q_j, D), \tag{1}$$

where s is a function specifying the score of candidate $a_{j,k}$ given question q_j and document D. $\phi_i(a_{j,k}, q_j, D)$ is a feature function, w_i is its corresponding weight, and D is a sequence of sentences. We describe the features in more detail in section 3.2, and the weight training algorithm in section 4

2.4 Answer Selection

The answer selection module chooses which, if any, question to answer. Let $a_{j,k}$ be a candidate answer for a given q_j and D, and let s_k be the score from evaluation $s(a_{j,k} \mid q, D)$. If a_{j,k_1} is the answer with the highest score among all candidates for q_j, we define the certainty of the answer to be

$$c_{j,k_1} = \min_{k_2 \neq k_1} (\mathrm{s}(a_{j,k_1} \mid q_j,\ D) - \mathrm{s}(a_{j,k_2} \mid q_j,\ D)) \tag{2}$$

[1] http://nlp.stanford.edu/software/CRF-NER.shtml
[2] http://www.lextek.com/manuals/onix/stopwords2.html

If our certainty exceeds a threshold t, we provide the answer a_{j,k_1} and if it does not, the system returns no answer.

As the main test data for this year evaluation contains a "none of the above" candidate answer for every question, we choose to answer 20% of the questions with this answer. This heuristic strategy is applied because our system is currently incapable to provide an analysis toward this type of negation and we do not have any gold-standard training data from previous years on which to perform training. Intuitively, we choose 20% questions with lowest score from Equation (1) to be assigned with the candidate answer "none of the above".

3 Model

3.1 Sentence-Matching Criterion

Our system uses a bags-of-n-grams vector space model for sentence matching. For example, if we have the sentence "We are scientists", we will have bags-of-words as follow: 1-gram = {"We", "are", "scientists"}, 2-gram = {"We are", "are scientists"}, 3-gram = {"We are scientists"}. The model that we used is a model consisting of bags-of-words for all n-grams and is defined as follows:

$$\text{model} = \text{1-gram} \cup \text{2-gram} \cup \text{3-gram}.$$

In preliminary experiments, we found $n = 3$ achieved higher precision than $n = 1, 2, 4, 5$, and that it was necessary to take the union of higher and lower order n-grams.

To measure similarity between vectors, we used TF/IDF [11] weighted cosine-similarity [5] as the weight of term i . Most of our features are based on the cosine similarity measure, which is commonly used in many IR systems

$$\text{tf-idf}(i, v) = \text{tf}(i, v) \times \log\left(\frac{|v|}{\text{n}(i, v)}\right), \tag{3}$$

where $\text{tf}(i, v)$ is the occurrence frequency of term i in passage v, $|v|$ is the total number of sentences in D, and $\text{n}(i, v)$ is the number of sentences in v in which term i occurs.

3.2 Features

Following from (3), the equation for cosine similarity between question q and sentence d_j is

$$\text{sim}(q, d_j) = \frac{\sum_{i=1}^{p} (\text{tf-idf}(i, q) \times \text{tf-idf}(i, d_j))}{\sqrt{\sum_{i=1}^{p} \text{tf-idf}(i, q)^2} \times \sqrt{\sum_{i=1}^{p} \text{tf-idf}(i, d_j)^2}}. \tag{4}$$

Each feature gives a score to each pair of $\{q, D\}$ and $\{a_k\}$. We use three intra-sentence features that are widely known in previous work:

1. Greatest Cosine (GC) finds the most related sentence to q and a_k. If this value is high, we have more certainty in the candidate answer. This feature concatenates the question and answer and finds the greatest cosine similarity value for a sentence in the document.Let p be a query where q and a_k are concatenated together.

$$\text{GC} \leftarrow \max_j(\text{sim}(p, d_j)).$$

2. Greatest Matching (GM) does not adopt cosine matching, but simply counts the maximum number of words that match both the a_k and q in a single sentence. This feature simply counts the greatest number of words overlapping between one sentence in the background text and the concatenated question and answer.

$$\text{GM} \leftarrow \max_j\ (|(q \cup a_k) \cap d_j|).$$

3. Cosine Matching (CosM) distinguishes whether the question and candidate answers occur in the same sentences in D or not. Here l is some threshold, which we set to 0.1 after preliminary tests.

$$\text{CosM} \leftarrow |D_1 \cap D_2| \text{ where } D_1 = \{d_j \mid \text{sim}(q, d_j) > l\} \wedge D_2 = \{d_j \mid \text{sim}(a_k, d_j) > l\}$$

We also propose new inter-sentence features that help capture answers that span multiple sentences in a simple manner:

1. Closest Matching (ClM) aims to find candidate answers that are not in the same sentence as q but close in proximity. We represent this using the distance from representative sentence r, which is defined as the sentence most similar to the question, and all answer candidate sentences. Let $\text{index}(d_j)$ be a function that indicates the location of d_j in the passage, $r = \underset{j}{\text{argmax}}(\text{sim}(q, d_j))$ and $D = \{d_j \mid \text{sim}(a_k, d_j) > l\}$.

$$\text{ClM} = \begin{cases} \min\limits_{d_j \in D}(|\text{index}(r) - \text{index}(d_j)|) & \text{if } D \neq \emptyset \\ 0 & \text{otherwise} \end{cases}$$

2. Closest Sentence (ClS) is quite similar to ClM, but instead of finding a representative sentence r, it counts the distance from all sentences that exceed the threshold t.
3. Unmatched (UM) is a binary feature active when ClM or ClS find no answer.

4 Weight Learning

This section describes a unified minimum error rate training [7] framework to learn *how* and *when* to answer questions by adjusting $\boldsymbol{w}$ and its scaling with respect to threshold t.

4.1 QA Evaluation Measures

Before learning, we must formally define how good any particular values of $\boldsymbol{w}$ are. One definition of the "goodness" of $\boldsymbol{w}$ is the *accuracy*, or percentage of questions answered correctly

$$\mathrm{a}(\boldsymbol{w}, t, Q) = \frac{\mathrm{c}(\boldsymbol{w}, t, Q)}{|Q|} \tag{5}$$

where $\mathrm{c}(\boldsymbol{w}, t, Q)$ is the number of questions in Q answered correctly given $\boldsymbol{w}$ and t. However, while this measure is intuitive, it also cannot distinguish between unanswered and incorrectly answered questions. As a remedy to this, [8] propose "c@1," which gives partial credit to unanswered questions, in proportion to the accuracy

$$\mathrm{c@1}(\boldsymbol{w}, t, Q) = \frac{\mathrm{c}(\boldsymbol{w}, t, Q) + \mathrm{n}(\boldsymbol{w}, t, Q)\mathrm{a}(\boldsymbol{w}, t, Q)}{|Q|} \tag{6}$$

where $\mathrm{n}(\boldsymbol{w}, t, Q)$ is the number of no-answers.

4.2 Minimum Error Rate Training

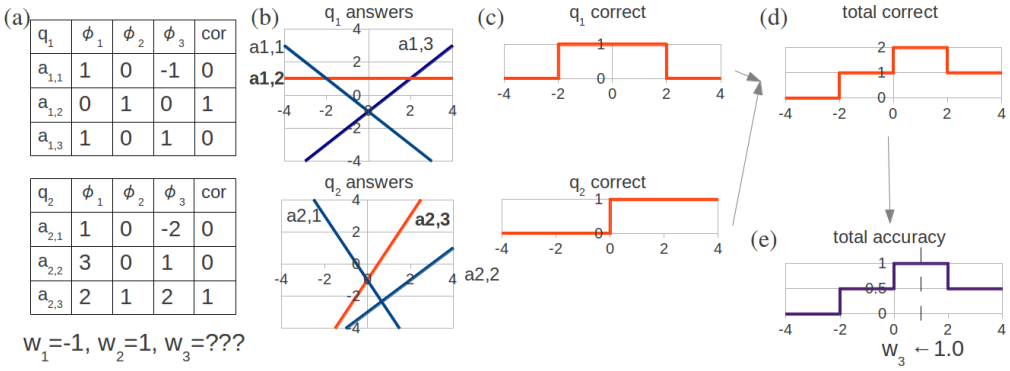

Fig. 2. Optimization of w_3: (a) training examples and initial $\boldsymbol{w}$, (b) slopes of each example, (c) whether q_j is correct for ranges of w_3, (d) total number of correct answers, (e) the accuracy and new value of w_3.

Next, we want to find a value of $\boldsymbol{w}$ that allows our system to score well on these evaluation measures. To do so, we first adopt the minimum error rate training (MERT) framework of [7], which can learn weights $\boldsymbol{w}$ for arbitrary evaluation measures that do not consider a threshold ($t = 0$).

The basic idea of MERT is to efficiently find weights that minimize some measure of error of the system. For example, we could define our error as one minus the accuracy of the system, and find weights that minimize this value

$$\hat{\boldsymbol{w}} = \operatorname*{argmin}_{\boldsymbol{w}} 1 - \mathrm{a}(\boldsymbol{w}, 0, Q) \tag{7}$$

As exactly solving Equation (7) is computationally difficult, [7] proposes an approximate algorithm using the coordinate ascent method of [10].[3] We will give a conceptual overview of the procedure here, and readers may refer to [6] or the supplementary code for a more complete algorithmic explanation. The basic idea of this method is that we iterate through each single element $w_i \in \boldsymbol{w}$, and find the value of w_i that minimizes error, given that all other elements of $\boldsymbol{w}$ (represented as $\boldsymbol{w} \backslash w_i$) are kept constant:

$$\hat{w_i} = \underset{w_i}{\operatorname{argmin}}\ 1 - \mathrm{a}(\boldsymbol{w}, 0, Q) \tag{8}$$

This process continues until no w_i can be modified to decrease the error rate.

Figure 2 shows this procedure on two questions with answers and their corresponding features in Figure 2 (a). First note that for answer $a_{j,k}$ to question q_j, Equation (1) can be decomposed into the part affected by w_i and the part affected by $\boldsymbol{w} \backslash w_i$:

$$s(q_j, a_{j,k}) = w_i \phi_i(q_j, a_{j,k}, D) + \sum_{h \neq i} w_h \phi_h(q_j, a_{j,k}, D) \tag{9}$$

Figure 2 (b) plots Equation (9) as a linear equation over w_i in the form $y = cx + d$ where $c = \phi_i(q_j, a_{j,k}, D)$, $x = w_i$ and d is equal to the right hand sum. These plots demonstrate for which values of w_i each answer $a_{j,k}$ will be assigned a particular score, with the highest line being the one chosen by the system. For q_1, the chosen answer will be $a_{1,1}$ for $w_i < -2$, to $a_{1,2}$ for $-2 < w_i < 2$, and to $a_{1,3}$ for $2 < w_i$ as indicated by the range where the answer's corresponding line is greater than the others.

Next, we take the information of answers chosen in Figure 2 (b) and, given the information about what answers are correct, convert this into a graph as Figure 2 (c) where the value is one for regions covered by the correct answer and zero otherwise. The ranges for each question are then combined into a single graph indicating the total number of correct answers for each value of w_i (Figure 2 (d)). Finally, given the statistics in Figure 2 (d), we can calculate the error function used in Equation (8) and choose a point in the center of the region with the minimal error as shown in Figure 2 (e).

4.3 Thresholded MERT

While the previous procedure can minimize errors that are merely concerned with the highest scoring answer, when considering a threshold t we also need to keep track of whether the best answer exceeds the second best answer by more than t. We present a modification to the standard MERT procedure that allows us to learn weights in the face of a threshold, which we will refer to as thresholded minimum error rate training (TMERT).[4]

[3] [4] present a method to improve the efficiency of exact MERT, but it is still significantly less efficient and more complicated than approximate solutions.

[4] Our implementations is available open source at `http://phontron.com/tmert`

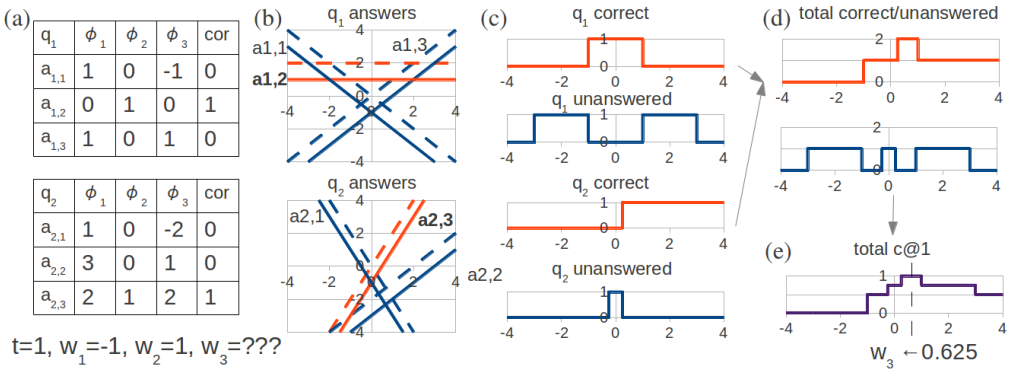

Fig. 3. Minimum error rate training with thresholds.

Figure 3 demonstrates our proposed algorithm, emphasizing parallels and differences to traditional MERT. The first difference is that for each *answer* line $y = cx + d$ in Figure 2 (b), we add an additional *threshold* line $y = cx + d + t$ raised by t. This encodes the information that if a particular answer line for a_{j,k_1} is less than or equal to the threshold line for a_{j,k_2}, a_{j,k_1} has failed to exceed a_{j,k_2} by t, and thus no answer should be returned. To find the response that will be provided by the system for any region in Figure 3 (b), we examine not the line with the highest score, but the line with the second highest score. For each range where the second highest line is an *answer* line for a_{j,k_1}, we know the first highest line is the corresponding threshold line for a_{j,k_1}, so a_{j,k_1} has exceeded all other answers by at least t and thus will be returned as the answer by the system. For each range where the second highest line is a *threshold* line, we can tell that the margin between the first and second answers is less than t, and thus the system will return no answer.

During the aggregation of statistics in Figure 3 (c), we collect information not only on regions where the answer was correct, but also on regions where the question was unanswered. Given these statistics, we can calculate accuracy for measures that reward non-response such as c@1 and choose a value of w_i that minimizes the error accordingly.

We can also indirectly optimize threshold t itself in this framework. To do so, we note that the overall scale of the weights $\boldsymbol{w}$ is inversely proportional to the threshold t; having large weights is equivalent to having a small threshold, and vice-versa. Thus, any $t > 0$ will have the same effect on our thresholded learning algorithm, so we simply set $t = 1$.[5]

[5] It is also possible to more efficiently adjust the scale of $\boldsymbol{w}$ by taking an additional TMERT optimization step for a scaling factor λ, then scale $\boldsymbol{w} \leftarrow \hat{\lambda}\boldsymbol{w}$.

5 Evaluation

5.1 Experimental Setup

For this year's QA4MRE, our system used only the English test set document and did not reference the background collection.

The main task consists of 284 questions, classified into 4 main topics ("AIDS", "Climate Change", "Music and Society" and "Alzheimer") and main task questions and documents that in standard and relatively simple language. There are 16 test documents (4 for each topic) and approximately 15–20 questions with 5 candidate answers for each test document. The system is required to choose answer from these multiple choices and only 1 correct candidate answer is available for each question. The system can leave the question answered if it lacks confidence. While the main task is quite similar to the past years, this year each question contains "none of the above" as a candidate answer.

To train the parameters of our model, we use both test set documents from past CLEF 2011 and 2012 QA4MRE campaigns [9]. As they provide 120 + 160 question with correct answers, we use this as our primary training data. We train the weights of the features using the described TMERT training algorithm. After training, the system achieved 112 correct answers, 15 answers with no candidate, and 153 wrong answers, yielding a c@1 score of 42% on the training data.

5.2 Main Task Results

We submit a total of 2 runs for the main task separating the results from different strategies. Run1 uses the strategy of choosing some answers as "none of the above" as mentioned in Section 2.4 while Run2 does not.

Topic	Run1		Run2	
	Main	+Aux	Main	+Aux
Alzheimer	0.36	0.36	0.29	0.29
Music and society	0.28	0.37	0.29	0.39
Climate Change	0.29	0.32	0.24	0.28
AIDS	0.19	0.25	0.15	0.23
Average C@1	0.28	**0.33**	0.24	0.30

Table 1. Result of Participation in Main Task

First, we show the results as measured by c@1 in Table 1. Our strategy in Run1 resulted in a gain in system performance for an additional 3–4% over Run2, for evaluation in both main and +Aux questions. It indicates that the strategy is able to slightly raise accuracy for our system. The gains are relatively stable among the topics, with the topic "Alzheimer" receiving the most benefit.

Overall, the system's best topic is "Alzheimer" and worst topic is "AIDS". The accuracies for the topic "Music and society" and "Climate change" are relatively similar. However as we did not use any form of knowledge except

reading the document itself, we cannot provide further analysis on why our system is good or bad at certain topic.

The results in Table 1 also indicates that the system is relatively weak at the main questions. In particular taking a look at the sample question r_id=5 and q_id=6:

> Of all Cramer's works, the one that has had the greatest enduring value is his celebrated set of 84 studies for the piano, published in two sets of 42 each in 1804 and 1810 as "Studio per il pianoforte". This collection has long been considered a cornerstone of pianistic technique and is the only work of Cramer's that is generally known today.
>
> Question: Why is the "Studio per il pianoforte" well known even today?
>
> 1. because there are 84 studies
> 2. because the studies are structurally simple
> 3. because it teaches piano technique very effectively
> 4. because it shows the influence of Scarlatti
> 5. none of the above

The system return the answer of "1" because there are many matching keywords of the candidate answer and question in the same sentence and our anaphora resolution module failed to match the word "This" as "Studio per il pianoforte". However, our inter-sentence features are also able to answer some questions that need an inter-sentence analysis (taken from r_id=7, q_id=9):

> The major Hollywood studios of the so-called Golden Age (c1935-55) were MGM, Paramount, RKO, Warner Brothers and 20th Century-Fox. Each housed a permanent music department, with contracted composers, arrangers, orchestrators, librarians and music editors, as well as a resident orchestra, all working under a senior music director.
>
> Question: What sort of music was written for Hollywood films in the Golden Age?
>
> 1. music for orchestra with strong melodies
> 2. music for singer and piano
> 3. music for youth audiences
> 4. music with four-track stereo sound
> 5. none of the above

This question was successfully answered with the help of the ClM features, which finds a distance of one between the question and candidate answer. The question matched the first sentence because the term "Golden Age" and "Hollywood" appeared in it and the next sentence matched candidate answer 1 because term "orchestra" appeared in it. So the system returned "1" as its final answer.

For the evaluation measure (7), the function c, and n are respectively the number of correct answers and no answers for paticular test set. Run1 achieved 88 correct answers, 182 wrong answers, and 14 unanswered, resulting a c@1 score of 32.51%.

6 Conclusion

As part of our participation in QA4MRE@CLEF 2013, we have developed QA-system that is simple but able to answer certain types of questions. In particular it achieved higher accuracy for simpler types of questions in the main task, but lacks in terms of answering more complex question types that need more sophisticated processing. For future work, we believe that it is necessary to use external knowledge such as background knowledge so the system can provide further analysis in classifying questions and determining certain type of strategies to answer the questions. Luckily our described framerwork for a linear combination of experts and the proposed MERT training metric are conducive to adding additional components to capture this information. Further work will be focussed on integrating external knowledge derived from sources such as Wikipedia and the background collections by adding more features.

References

1. Brill, E., Dumais, S., Banko, M.: An Analysis of the AskMSR Question-Answering System. In: In Proceedings of EMNLP. pp. 257–264 (2002)
2. Chu-Carroll, J., Czuba, K., Prager, J., Ittycheriah, A.: In Question Answering, Two Heads Are Better Than One. In: In HLT-NAACL. pp. 24–31 (2003)
3. Dang, H.T., Lin, J.J., Kelly, D.: Overview of the TREC 2006 Question Answering Track 99. In: TREC (2006)
4. Galley, M., Quirk, C.: Optimal Search for Minimum Error Rate Training. In: Proceedings of EMNLP. pp. 38–49 (2011)
5. Jurafsky, D., Martin, J.H.: Speech and Language Processing. Pearson Education, Upper Saddle River, NJ, 2 edn. (2008)
6. Koehn, P.: Section 9.3: Parameter tuning. In: Statistical Machine Translation, pp. 263–271. Cambridge Press (2010)
7. Och, F.J.: Minimum error rate training in statistical machine translation. In: Proceedings of ACL (2003)
8. Peñas, A., Rodrigo, A.: A Simple Measure to Assess Non-response. In: Proceedings of ACL. pp. 1415–1424. Association for Computational Linguistics, Portland, Oregon, USA (June 2011)
9. Peñas, A., Hovy, E.H., Forner, P., Rodrigo, Á., Sutcliffe, R.F.E., Forascu, C., Sporleder, C.: Overview of QA4MRE at CLEF 2011: Question Answering for Machine Reading Evaluation. In: Petras, V., Forner, P., Clough, P.D. (eds.) CLEF (Notebook Papers/Labs/Workshop) (2011)
10. Powell, M.: An efficient method for finding the minimum of a function of several variables without calculating derivatives. The Computer Journal 7(2), 155–162 (1964)
11. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. In: Information Processing and Management. pp. 513–523 (1988)
12. Tufis, D.: Natural Language Question Answering in Open Domains. The Computer Science Journal of Moldova 19(2), 146–164 (2011)