Full Machine Translation for Factoid Question Answering

Cristina España-Bonet and **Pere R. Comas**
TALP Research Center
Universitat Politècnica de Catalunya (UPC)
{cristinae,pcomas}@lsi.upc.edu

Abstract

In this paper we present an SMT-based approach to Question Answering (QA). QA is the task of extracting exact answers in response to natural language questions. In our approach, the answer is a translation of the question obtained with an SMT system. We use the n-best translations of a given question to find similar sentences in the document collection that contain the real answer. Although it is not the first time that SMT inspires a QA system, it is the first approach that uses a full Machine Translation system for generating answers. Our approach is validated with the datasets of the TREC QA evaluation.

1 Introduction

Question Answering (QA) is the task of extracting short, relevant textual answers from a given document collection in response to natural language questions. QA extends IR techniques because it outputs concrete answers to a question instead of references to full documents which are relevant to a query. QA has attracted the attention of researchers for some years, and several public evaluations have been recently carried in the TREC, CLEF, and NTCIR conferences (Dang et al., 2007; Peñas et al., 2011; Sakai et al., 2008). All the example questions of this paper are extracted from the TREC evaluations.

QA systems are usually classified according to what kind of questions they can answer; *factoid*, *definitional*, *how to* or *why* questions are treated in a distinct way. This work focuses on *factoid* questions, that is, those questions whose answers are semantic entities (e.g., organisation names, person names, numbers, dates, objects, etc.). For example, the question *Q1545: What is a female rabbit called?* is factoid and its answer, *"doe",* is a semantic entity (although not a named entity).

Factoid questions written in natural language contain implicit information about the relations between the concepts expressed and the expected outcomes of the search, and QA explicitly exploits this information. Using an IR engine to look up a boolean query would not consider the relations therefore losing important information. Consider the question *Q0677: What was the name of the television show, starring Karl Malden, that had San Francisco in the title?* and the candidate answer A. In this question, two types of constraints are expressed over the candidate answers. One is that the expected type of A is a kind of "television show." The rest of the question indicates that "Karl Malden" is related to A as being "starred" by, and that "San Francisco" is a substring of A. Many factoid questions explicitly express an hyponymy relation about the answer type, and also several other relations describing its context (i.e. spatial, temporal, etc.).

The QA problem can be approached from several points of view, ranging from simple surface pattern matching (Ravichandran and Hovy, 2002), to automated reasoning (Moldovan et al., 2007) or supercomputing (Ferrucci et al., 2010). In this work, we propose to use Statistical Machine Translation (SMT) for the task of factoid QA. Under this perspective, the answer is a translation of the question. It is not the first time that SMT is used for QA tasks, several works have been using translation models to determine the answers (Berger et al., 2000; Cui et al., 2005; Surdeanu et al., 2011). But to our knowledge this is the first approach that uses a full Machine Translation system for generating answers.

The paper is organised as follows: Section 2 reviews the previous usages of SMT in QA, Section 3 reports our theoretical approach to the task, Section 4 describes our QA system, Section 5 presents the experimental setting, Section 6 analyses the results and Section 7 draws conclusions.

2 Translation Models in QA

The use of machine translation in IR is not new. Berger and Lafferty (1999) firstly propose a probabilistic approach to IR based on methods of SMT. Under their perspective, the human user has an information need that is satisfied by an "ideal" theoretical document d from which the user draws important query words q. This process can be mirrored by a translation model: given the query q, they find the documents in the collection with words a most likely to translate to q. The key ingredient is the set of translation probabilities $p(q|a)$ from IBM model 1 (Brown et al., 1993).

In a posterior work, Berger et al. also introduce the formulation of the QA problem in terms of SMT (Berger et al., 2000). They estimate the likelihood that a given answer containing a word a_i corresponds to a question containing word q_j. This estimation relies on an IBM model 1. The method is tested with a collection of closed-domain Usenet and call-center questions, where each question must be paired with one of the recorded answers. Soricut and Brill (2004) implement a similar strategy but with a richer formulation and targeted to open-domain QA. Given a question Q, a web-search engine is used to retrieve 3-sentence-long answer texts from FAQ pages. These texts are later ranked with the likelihood of containing the answer to Q, and this likelihood is estimated via a noisy-channel architecture. The work of Murdock and Croft (2005) applies the same strategy to TREC data. They evaluate the TREC 2003 passage retrieval task. In this task, the system must output a single sentence containing the answer to a factoid question. Murdock and Croft tackle the length disparity in question-answer pairs and show that this MT-based approach outperforms traditional query likelihood techniques.

Riezler et al. (2007) define the problem of answer retrieval from FAQ and social Q/A websites as a query expansion problem. SMT is used to translate the original query terms to the language of the answers, thus obtaining an expanded list of terms usable in standard IR techniques. They also use SMT to perform question paraphrasing. In the same context, Lee et al. (2008) study methods for improving the translation quality removing noise from the parallel corpus.

SMT can be also applied to sentence representations different than words. Cui et al. (2005) approach the task of passage retrieval for QA with translations of dependency parsing relations. They extract the sequences of relations that link each pair of words in the question and, using the IBM translation model 1, score their similarity to the relations extracted from the candidate passage. Thus, an approximate relation matching score is obtained. Surdeanu et al. (2011) extend the scope of this approach by combining together the translation probabilities of words, dependency relations, and semantic roles in the context of answer searching in FAQ collections.

The works we have described so far use archives of question-answer pairs as information sources. They are really doing document retrieval and sentence retrieval rather than question answering, because every document/sentence is known to be the answer of a question written in the form of an answer, and no further information extraction is necessary, they just select the best answer from a given pool of answers. The difference with a standard IR task is that these systems are not searching for *relevant* documents but for *answer* documents. In contrast, Echihabi and Marcu (2003) introduce an SMT-based method for extracting the concrete answer in factoid QA. First, they use a standard IR engine to retrieve candidate sentences and process them with a constituent parser. Then, an elaborated process simplifies these parse trees converting them into sequences of relevant words and/or syntactic tags. This process reduces the length disparity between questions and answers. For the answer extraction, a special tag marking the position of the answer is sequentially added to all suitable positions in the sentence, thus yielding several candidate answers for each sentence. Finally, each answer is rated according to its likelihood of being a translation of the question, according to an IBM model 4 trained on a corpus of TREC and web-based question-answer pairs.

With the exception of the query expansion approaches (Riezler et al., 2007), all works discussed here use some form of noisy-channel model (translation model and target language model) but do not perform the decoding part of the SMT process to generate translations, nor use the rich set of features of a full SMT. In fact, the formulation of the noisy-channel in these works has very few differences with pure language modelling approaches to QA like the one of Heie et al. (2011), where two different models for retrieval and filtering are learnt from a corpus of question-answer pairs.

3 Question-to-Answer Translation

The core of our QA system is an SMT system for the Question-to-Answer language pair. In SMT, the best translation for a given source sentence is the most probable one, and the probability of each translation is given by the Bayes theorem. In our case, the source sentence corresponds to the question Q and the target or translation is the sentence containing the answer A. With this correspondence, the fundamental equation of SMT can be written as:

$$\begin{aligned}\mathcal{A}(Q) &= \hat{A} = \mathrm{argmax}_A\, P(A|Q) \\ &= \mathrm{argmax}_A\, P(Q|A)\, P(A), \qquad (1)\end{aligned}$$

where $P(Q|A)$ is the translation model and $P(A)$ is the language model, and each of them can be understood as the sum of the probabilities for each of the segments or phrases that conform the sentence. The translation model quantifies the appropriateness of each segment of Q being answered by A; the language model is a measure of the fluency of the answer sentence and does not take into account which is the question. Since we are interested in identifying the concrete string that answers the question and not a full sentence, this probability is not as important as it is in the translation problem.

The log-linear model (Och and Ney, 2002), a generalisation of the original noisy-channel approach (Eq. 1), estimates the final probability as the logarithmic sum of several terms that depend on both the question Q and the answer sentence A. Using just two of the features, the model reproduces the noisy-channel approach but written in this way one can include as many features as desired at the cost of introducing the same number of free parameters. The model in its traditional form includes 8 terms:

$$\begin{aligned}\mathcal{A}(Q) &= \hat{A} = \mathrm{argmax}_{\mathrm{A}}\, \log P(A|Q) = \\ &+ \lambda_{lm} \log P(A) + \lambda_d \log P_d(A,Q) \\ &+ \lambda_{lg} \log lex(Q|A) + \lambda_{ld} \log lex(A|Q) \\ &+ \lambda_g \log P_t(Q|A) + \lambda_d \log P_t(A|Q) \\ &+ \lambda_{ph} \log ph(A) + \lambda_w \log w(A)\,, \qquad (2)\end{aligned}$$

where $P(A)$ is the language model probability, $lex(Q|A)$ and $lex(A|Q)$ are the generative and discriminative lexical translation probabilities respectively, $P_t(Q|A)$ the generative translation model, $P_t(A|Q)$ the discriminative one, $P_d(A,Q)$ the distortion model, and $ph(A)$ and $w(A)$ correspond to the phrase and word penalty models. We start by using this form for the answer probability and analyse the importance and validity of the terms in the experiments Section. The λ weights, which account for the relative importance of each feature in the log-linear probabilistic model, are commonly estimated by optimising the translation performance on a development set. For this optimisation one may use Minimum Error Rate Training (MERT) (Och, 2003) where BLEU (Papineni et al., 2002) is the reference evaluation.

Once the weights are determined and the probabilities estimated from a corpus of question-answer pairs (a parallel corpus in this task), a decoder uses Eq. 2 to score the possible outputs and to find the best answer sentence given a question or, in general, an n-best list of answers.

This formulation, although possible from an abstract point of view, is not feasible in practice. The corpus from which probabilities are estimated is finite, and therefore new questions may not be represented. There is no chance that SMT can generate *ex nihilo* the knowledge necessary to answer questions such as *Q1201: What planet has the strongest magnetic field of all the planets?*. So, rather than generating answers via translation, we use translations as indicators of the sentence *context* where an answer can be found. Context here has not only the meaning of near words but also a context at a higher level of abstraction.

To achieve this, we use two different representations of the question-answer pairs and two different SMT models in our QA system. We call Level1 representation the original strings of text of the question-answer pairs. The Level2 representation, that aims at being more abstract, more general and more useful in SMT, is constructed applying this sequence of transformations: 1) Quoted expressions in the question are identified, paired with their counterpart in the answer (in case any exists) and substituted by a special tag QUOTED. 2) Each named entity is substituted by its entity class (e.g., “Karl Malone” by PERSON). 3) Each noun and verb is substituted by their WordNet supersense[1] (e.g. “nickname” by COMMUNICATION). 4) Any remaining word, such as adjectives, adverbs and stop words, is left as is. Additionally, in the answer sentence string, the correct answer entity is substituted by a special tag ANSWER. An example of this annotation is given in Figure 1.

An SMT system trained with Level1 examples will translate Q to answer sentences with vocabulary and structure similar to the learning examples. The Level2 system will translate to a mix of named entities, WordNet supersenses, bare words, and ANSWER markers that represent the abstract structure of the answer sentence. We call *patterns* to the Level2 translations. The rationale of this process is that the SMT model can learn the context where answers appear depending of the structure of the question. The obtained translations from both levels can be searched in the document collection to find sentences that are very similar.

Note that in Level2, the vocabulary size of the question-answer pairs is dramatically reduced with respect to the original Level1 sentences, as seen in Table 2. Thus, the sparseness is reduced, and the translation model gains in coverage; patterns are also easier to find than Level1 sentences, and give flexibility and generality to the translation. And the most important feature, patterns capture the context of the answer, pinpointing it with accuracy.

These Level1 and Level2 translations are the core of our QA system that is presented in the following Section.

4 The Question Answering System

Our QA system is a pipeline of three modules. In the first one, the question is analysed and annotated with several linguistic processors. This information is used by the rest of the modules. In the second one, relevant documents are obtained from the document collection with straightforward IR techniques and a list of candidate answers is generated. Finally, these candidate answers are filtered and ranked to obtain a final list of proposed answers. This pipeline is a common architecture for a simple QA system.

Level1 Q: What is Karl Malone’s nickname ?
Level1 A: Malone , whose overall consistency has earned him the nickname `ANSWER` , missed both of them with nine seconds remaining .

Level2 Q: What STATIVE B-PERSON ’s COMMUNICATION ?
Level2 A: B-PERSON , whose overall ATTRIBUTE POSSESSION POSSESSION him the COMMUNICATION `ANSWER` , PERCEPTION both of them with B-NUM TIME CHANGE .

Figure 1: Example of the two annotation levels used.

4.1 Question Analysis

Questions are processed with a tokeniser, a POS tagger, a chunker, and a NERC. Besides, each word is tagged with its most frequent sense in WordNet. Then, a maximum-entropy classifier determines the most probable expected answer types for the question (EAT). This classifier is built following the approach of Li and Roth (2005), it can classify questions into 53 different answer types and belongs to our in-house QA system. Finally, a weighted list of relevant keywords is extracted from the question. Their saliences are heuristically determined: the most salient tokens are the quoted expressions, followed by named entities, then sequences of nouns and adjectives, then nouns, and finally verbs and any remaining non-stop word. This list is used in the candidate answer generation module.

4.2 Candidate Answer Generation

The candidate answer generation comprises two steps. First a set of passages is retrieved from the document collection, and then the candidate answers are extracted from the text.

For the retrieval, we have used the passage retrieval module of our in-house QA system. The passage retrieval algorithm initially creates a boolean query with all nouns and more salient words, and sets a threshold t to 50. It uses the Lucene IR engine[2] to fetch the documents match-

[1] WordNet noun synsets are organised in 26 semantic categories based on logical groupings, e.g., ARTIFACT, ANIMAL, BODY, COMMUNICATION…The verbs are organised in 15 categories. (Fellbaum, 1998)

[2] `http://lucene.apache.org`

ing the current query and a subsequent passage construction module extracts passages as document segments where two consecutive keyword occurrences are separated by at most t words. If too few or too many passages are obtained this way, a relaxation procedure is applied. The process iteratively adjusts the salience level of the keywords used in the query by dropping low salient words when too few are obtained or adding them when too many, and it also adjusts their proximity threshold until the quality of the recovered information is satisfactory (see Surdeanu et al. (2006) for further details).

When the passages have been gathered, they are split into sentences and processed with POS tagging, chunking and a NERC. The candidate answer list is composed of all named entities and all phrases containing a noun. Each candidate is associated to the sentence it has been extracted from.

4.3 Answer Ranking

This module selects the best answers from the candidates previously generated. It employs three families of scores to rank them.

Context scores B and R: The n-best list of Level2 question translations is generated. In this step retrieved sentences are also transformed to the Level2 representation. Then, each candidate answer is replaced by the special ANSWER tag in the associated sentence, thus, each sentence has a unique ANSWER tag, as in the training examples. Finally, each candidate is evaluated assessing the similarity of the source sentence with the n-best translations.

For this assessment we use two different metrics. One of them is a lexical metric commonly used in machine translation, BLEU (Papineni et al., 2002). A smoothed version is used to evaluate the pairs at sentence level yielding the score B. The other metric is ROUGE (Lin and Och, 2004), here named R. We use the skip-bigram overlapping measure with a maximum skip distance of 4 unigrams (ROUGE-S4). Contrary to BLEU, ROUGE-S does not require consecutive matches but is still sensitive to word order.

Both BLEU and ROUGE are well-known metrics that are useful for finding partial matchings in long strings of words. Therefore it is an easy way of implementing an approximated pattern matching algorithm with off-the-shelf components.

Although these scores can determine if a sentence is a candidate for asserting a certain property of a certain object, they do not have the power to discriminate if these objects are the actually required by the question. Level2 representation is very coarse and, for example, treats all named entities of the same categories as the same word. Thus, it is prone to introduce noise in the form of totally irrelevant answers. For example, consider the questions *Q1760: Where was C.S. Lewis born?* and *Q1519: Where was Hans Christian Anderson born?*. Both questions have the same Level2 representation: *Where STATIVE PERSON STATIVE?*, and the same n-best list of translations. Any sentence stating the birthplace (or even deathplace) of any person is equally likely to be the correct answer of both questions because the lexicalisation of Lewis and Anderson is lost.

On the other hand, B and R also show another limitation. Since they are based on n-gram matching, they cannot be discriminative enough when there is only one different token between options, and that happens when a same sentence has different candidates for the answer. In this case the system would be able to distinguish among answer sentences but then all the variations with the answer in a different position would have too much similar scores. In order to mitigate these drawbacks, we consider two other scores.

Language scores L$_b$, L$_r$, L$_f$: To alleviate the discriminative problem of the context matching metrics, we calculate the same B and R scores but with Level1 translations and the original lexicalised question. These are the L$_b$ and L$_r$ scores.

Additionally, we introduce a new score L$_f$ that does not take into account the n-gram structure of the sentences: after the n-best list of Level1 question translations is generated, the frequency of each word present in the translations is computed. Then, the words in the candidate answer sentence are scored according to their normalised frequency in the translations list and added up together. This score lies in the $[0, 1]$ range.

Expected answer type score E: This score checks if the type of the answer we are evaluating matches the expected types we have determined in the question analysis. For this task, the expected answer types are mapped to named entities and/or supersenses (e.g., type ENTY:product is mapped to ARTIFACT). If the candidate answer is a named entity of the expected type, or contains a noun of the expected supersense, then this candidate receives a score E equal to the confidence of the question classification (the scores of the ME classifier have been previously normalised to probabilities).

These three families of scores can be combined in several ways in order to produce a ranked list of answers. In Section 6 the combination methods are discussed.

5 Experiments

5.1 Training and Test Corpora

We have used the datasets from the Question Answering Track of the TREC evaluation campaigns[3] ranging from TREC-9 to TREC-16 in our experiments. These datasets provide both a robust testbed for evaluation, and a source of question-answer pairs to use as a parallel corpus for training our SMT system. Each TREC evaluation provides a collection of documents composed of newspaper texts (three different collections have been used over the years), a set of new questions, and an answer key providing both the answer string and the source document. Description of these collections can be found in the TREC overviews (Voorhees, 2002; Dang et al., 2007).

We use the TREC-11 questions for test purposes, the remaining sets are used for training unless some parts of TREC-9, TREC-10 and TREC-12 that are kept for fitting the weights of our SMT system. To gather the SMT corpus, we select all the factoid questions whose answer can be found in the documents and extract the full sentence that contains the answer. With this methodology, a parallel corpus with 12,335 question-answer pairs is obtained. We have divided it into two subsets: the pairs with only a single answer found in the documents are used for the development set, and the remaining pairs (i.e. having multiple occurrences of the correct answer) are used for training. The test set are the 500 TREC-11 questions, 452 out of them have a correct answer in the documents. The numbers are summarised in Table 1.

In order to obtain the Level2 representation of these corpora, the documents and the test sets must be annotated. For the annotation pipeline we use the TnT POS tagger (Brants, 2000), WordNet (Fellbaum, 1998), the YamCha chunker (Kudo and Matsumoto, 2003), the Stanford NERC (Finkel et al., 2005), and an in-house temporal expressions recogniser.

	Q	A	TRECs
Train	2264	12116	9,10,12,13,14,15,16
Dev	219	219	9,10,12
Test	500	2551	11

Table 1: Number of Questions and Answers in our data sets. The number of TREC evaluation from which are obtained is indicated.

	Tokens		Vocabulary	
	Q	A	Q	A
TrainL1	97028	393978	3232	32013
TrainL2	91567	373008	540	9130

Table 2: Statistics for the 12,116 Q-A pairs in the training corpus according to the annotation level.

Table 2 shows some statistics for the parallel corpus and the two different levels of annotation. From the SMT point of view the corpus is small in order to estimate the translation probabilities in a reliable way but, as stated before, Level2 representation diminishes the vocabulary considerably and alleviates the problem.

5.2 SMT system

The statistical system is a state-of-the-art phrase-based SMT system trained on the previously introduced corpus. Its development has been done using standard freely available software. The language model is estimated using interpolated Kneser-Ney discounting with SRILM (Stolcke, 2002). Word alignment is done with GIZA++ (Och and Ney, 2003) and both phrase extraction and decoding are done with the Moses package (Koehn et al., 2007). The model weights are optimised with Moses' script of MERT against the BLEU evaluation metric.

For the full model, we consider the language model, direct and inverse phrase probabilities, direct and inverse lexical probabilities, phrase and word penalties, and a non-lexicalised reordering.

5.3 QA system

The question answering system has three different modules as explained in Section 4. For the

[3] http://trec.nist.gov/data/qamain.html

	T1	T50	MRR
QA	0.006 (4)	0.206 (14)	0.024 (4)
SR	0.066 (8)	0.538 (9)	0.142 (8)
Upper bound	0.677	0.677	0.677

Table 3: Mean and standard deviation for 1000 realisations of the random baseline for QA and SR. The upper bound is also shown.

first module, questions are annotated using the same tools introduced in the corpora Section. The second module generates 2,866,098 candidate answers (373,323 different sentences), that is to say, a mean of 5,700 answers per question (750 sentences per question). These candidates are made available to the third module resulting in the experiments that will be discussed in Section 6.

The global QA system performance is evaluated with three measures. T1 is a measure of the system's precision and gives the percentage of correct answers in the first position; T50 gives the number of correct answers in the first 50 positions, in some cases that corresponds to all candidate answers; finally the Mean Reciprocal Rank (MRR) is a measure of the ranking capability of the system and is estimated as the mean of the inverse ranking of the first correct answer for every question: MRR= $Q^{-1} \sum_i \mathrm{rank}_i^{-1}$.

6 Results Analysis

Given the set of answers retrieved by the candidate answer generation module, a naïve baseline system is estimated by selecting randomly 50 answers for each of the questions. Table 3 shows the mean of the three measures after applying this random process 1000 times. The upper bound of this task is the oracle that selects always the correct answer/sentence if it is present in the retrieved passages. An answer is considered correct if it perfectly matches the official TREC's answer key and a sentence is correct if it contains a correct answer. The random baseline has a precision of 0.6%.

We also evaluate a second task, sentence retrieval for QA (SR). In this task, the system has to provide a sentence that contains the answer, but not to extract it. Within our SMT approach, both tasks are done simultaneously, because the answer is extracted according to its context sentence. A random baseline for this second task, where only full sentences without marked answers are taken into account, can also be read in Table 3.

	QA			SR		
Metric	T1	T50	MRR	T1	T50	MRR
B	0.018	0.292	0.049	0.084	0.540	0.164
R	0.018	0.283	0.045	0.119	0.608	0.209
B+R	0.022	0.294	0.053	0.097	0.573	0.180
BR	0.027	0.294	0.057	0.137	0.591	0.211

Table 4: System performance using an SMT that generates a 100-best list, uses a 5-gram LM and all the features of the TM.

1st best: The B-ORGANIZATION B-LOCATION , B-DATE (B-ORGANIZATION) - B-PERSON , whose COMMUNICATION STATIVE " ANSWER . "

50th best: The ANSWER ANSWER , B-DATE (B-ORGANIZATION) - B-PERSON , the PERSON of ANSWER , the most popular ARTIFACT , serenely COGNITION COMMUNICATION .

100th best: The B-LOCATION , B-DATE (B-ORGANIZATION) - B-PERSON , the PERSON of ANSWER , COMMUNICATION B-LOCATION 's COMMUNICATION .

Figure 2: Example of patterns found in an n-best list.

We begin this analysis studying the performance of the SMT-based parts alone. Table 4 shows the results when using an SMT decoder that generates a 100-best list, uses a 5-gram language model and all the features of the translation model. An example of the generated patterns in Level2 representation can be found in Figure 2 for the question of Figure 1, *Q1565: What is Karl Malone's nickname?*.

Candidate answer sentences are ranked according to the similarity with the patterns generated by translation as measured by BLEU (B), ROUGE-S4 (R) or combinations of them. To calculate these metrics the n-best list with patterns is considered to be a list of reference translations (Fig. 2) to every candidate (Fig. 1). In general, a combination of both metrics is more powerful than any of them alone and the product outperforms the sum given that in most cases BLEU is larger than ROUGE and smooths its effect. The inclusion of the SMT patterns improves the baseline but it does not imply a quantum leap. T1 is at least three times better than the baseline's one but still the system answers less than a 3% of the questions. In the first 50 positions the answer is

SMT Features	T1	T50	MRR
Lex, LM5, 100-best	0.027	0.294	0.057
noLex, LM5, 100-best	0.015	0.281	0.045
Lex, LM3, 100-best	0.015	0.257	0.041
Lex, LM7, 100-best	0.033	0.288	0.050
Lex, LM5, 10-best	0.024	0.310	0.056
Lex, LM5, 1000-best	0.027	0.301	0.061
Lex, LM5, 10000-best	0.011	0.290	0.045

Table 5: System performance with different combinations of the SMT features used in decoding. BR is the metric used to score the answers.

found a 30% of the times. In the sentence retrieval task, results grow up to 14% and 59% respectively. Its difference between tasks shows one of the limitations of these metrics commented before, they are not discriminative enough when the only difference among options is the position of the ANSWER tag inside the sentence. This is the empirical indication of the need for a score like E. On the other hand, each question has a mean of 5,732 candidate answers, and although T50 is not a significant measure, its good results indicate that the context scores metrics are doing their job. The highest T50, 0.608, is reached by R and it is very close to the upper bound 0.667.

Taking BR as a reference measure, we investigate the impact of three features of the SMT in Table 5. Regarding the length of the language model used in the statistical translation, there is a trend to improve the accuracy with longer language models (T1 is 0.015 for a LM3, 0.027 for LM5 and 0.033 for LM7 with the product of metrics) but recall is not very much affected and the best values are obtained for LM5.

Second, the number of features in the translation model indicates that the best scores are reached when one reproduces the same number of features as a standard translation system. That is, all of the measures when the lexical translation probabilities are ignored are significantly lower than when the eight features are used. In a counterintuitive way, the token to token translation probability helps to improve the final system although word alignments here can be meaningless or nonexistent given the difference in length and structure between question and answer.

Finally, the length of the n-best list is not a decisive factor to take into account. Since the elements in a n-best list usually differ very little, and this is even more important for a system with a reduced vocabulary, increasing the size of the list does not enrich in a substantial way the variety of the generated answers and results show no significant variances. Given these observations, we fix an SMT system with a 5-gram language model, the full set of translation model features and the generation of a 100-best list for obtaining B and R scores.

	QA			SR		
Metric	T1	T50	MRR	T1	T50	MRR
L$_f$	0.016	0.286	0.046	0.137	0.605	0.236
L$_b$	0.022	0.304	0.054	0.100	0.581	0.192
L$_r$	0.018	0.326	0.060	0.131	0.627	0.225
L$_{brf}$	0.038	0.330	0.079	0.147	0.622	0.238
E	0.044	0.373	0.096	0.058	0.579	0.142
EL$_{brf}$	0.018	0.293	0.048	0.118	0.623	0.214
BL$_{brf}$	0.051	0.337	0.091	0.184	0.616	0.271
RL$_{brf}$	0.033	0.346	0.069	0.191	0.618	0.279
BRL$_{brf}$	0.042	0.350	0.082	0.182	0.616	0.273
(B+R)L$_{brf}$	0.044	0.346	0.085	0.187	0.618	0.273
BE	0.035	0.384	0.084	0.086	0.579	0.179
RE	0.035	0.377	0.086	0.131	**0.630**	0.228
BRE	0.049	0.377	0.098	0.135	0.608	0.220
(B+R)E	0.040	**0.386**	0.091	0.102	0.596	0.196
BEL$_{brf}$	0.093	0.379	0.137	0.200	0.621	0.283
REL$_{brf}$	0.071	0.377	0.123	**0.208**	0.619	**0.294**
BREL$_{brf}$	0.091	0.379	0.132	0.200	0.622	0.287
(B+R)EL$_{brf}$	**0.100**	0.377	**0.141**	0.204	0.621	0.286

Table 6: System performance according to three different ranking strategies: context score (B and R), the language scores (L$_x$) and EAT type checking (E).

Each score approaches different problems of the task and therefore, complement each other rather than overlapping. Table 6 introduces the results of a selected group of score combinations, where L$_{brf}$=L$_b$L$_r$L$_f$.

The scores L$_{brf}$ and E alone are not very useful because L$_{brf}$ gives the same score to all candidates in the same sentence and E gives the same score to all candidates of the same type. Experimental results confirm that, as expected, L$_{brf}$ is more appropriate for the SR task and E for the QA task, although the figures are very low. When joining E and the Ls together, no improvement is obtained, and the results for the QA task are worse than L$_{brf}$ alone, thus demonstrating that Level1 translations are not good enough for the QA task.

A better system combines all the metrics together.

The best results are achieved when adding B and R scores to the combination. All of these combinations (i.e. B, R, BR and B+R) are better when are multiplied by both E and L_{brf} than by only one of them alone. Otherwise, combinations of only E and L_{brf} yield very poor results. Thus, the Level2 representation boosts T1 scores from 0.018 (EL_{brf}) to 0.100 ((B+R)EL_{brf}) in QA and almost doubles it in SR. As a general trend, we see that combinations involving R but not B are better in the SR task than in the QA task. In fact the best results for SR are obtained with the REL_{brf} combination. The best MRR scores are achieved also with the best T1 scores.

7 Discussion and Conclusions

The results here presented are our approach to consider question answering a translation problem. Questions in an abstract representation (Level2) are translated into an abstract representation of the answer, and these generated answers are matched against all the candidates obtained with the retrieval module. The candidates are then ranked according to their similarity with the n-best list of translations as measured by three families of metrics that include R, B, E and L.

The best combination of metrics is able to answer a 10.0% of the questions in first place (T1). This result is in the lowest part of the table reported by the official TREC-11 overview (Voorhees, 2002). The approach of Echihabi and Marcu (2003) that uses translation probabilities to rank the answers achieves higher results on the same data set (an MRR of 0.325 versus our 0.141). Although both works use SMT techniques, the approach is quite different. In fact, our system is more similar in spirit to that of Ravichandran and Hovy (2002), which learns regular expressions to find answer contexts and shows significant improvements for out-of-domain test sets, that is web data. Besides the fact that Echihabi and Marcu use translation models instead of a full translation system, they explicitly treat the problem of the difference of length between the question and the answer. In our work, this is not further considered than by the word and phrase penalty features of the translation model. Future work will address this difficulty.

The results of sentence ranking of our system are similar to those obtained by Murdock and Croft (2005), however, since test sets are different they are not directly comparable. This is notable because we tackle QA, and sentence retrieval is obtained as collateral information.

Possible lines of future research include the study abstraction levels different from Level2. The linguistic processors provide us with intermediate information such as POS that is not currently used as it is WordNet and named entities. Several other levels combining this information can be also tested in order to find the most appropriate degree of abstraction for each kind of word.

The development part of the SMT system is a delicate issue. MERT is currently optimising towards BLEU, but the final score for ranking the answers is a combination of a smoothed BLEU, ROUGE, L and E. It has been shown that optimising towards the same metric used to evaluate the system is beneficial for translation, but also that BLEU is one of the most robust metrics to be used (Cer et al., 2010), so the issue has to be investigated for the QA problem. Also, refining BLEU and ROUGE for this specific problem can be useful. A first approximation could be an adaptation of the n-gram counting of BLEU and ROUGE so that it is weighted by its distance to the answer; this way sentences that differ only because of the candidate answer string would be better differentiated.

Related to this, the generation of the candidate answer strings is exhaustive; the suppression of the less frequent candidates could help to eliminate noise in the form of irrelevant answer sentences. Besides, the system correlates these answer strings with the expected answer type of the question (coincidence measured with E). This step should be replaced by an SMT-based mechanism to build a full system only based on SMT. Furthermore, we plan to include the Level1 translations into the candidate answer generation module in order to do query expansion in the style of Riezler et al. (2007).

Acknowledgements

This work has been partially funded by the European Community's Seventh Framework Programme (MOLTO project, FP7-ICT-2009-4-247914) and the Spanish Ministry of Science and Innovation projects (OpenMT-2, TIN2009-14675-C03-01 and KNOW-2, TIN2009-14715-C04-04).

References

A. Berger and J. Lafferty. 1999. Information retrieval as statistical translation. In *Proceedings of ACM SIGIR Conference*.

A. Berger, R. Caruana, D. Cohn, D. Freitag, and V. Mittal. 2000. Bridging the lexical chasm: statistical approaches to answer-finding. In *Proceedings of the ACM SIGIR Conference*.

T. Brants. 2000. TnT – a statistical part-of-speech tagger. In *Proceedings ANLP Conference*.

P. F. Brown, V. J. Della Pietra, S. A. Della Pietra, and R. L. Mercer. 1993. The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*.

D. Cer, C. D. Manning, and D. Jurafsky. 2010. The best lexical metric for phrase-based statistical MT system optimization. In *Proceeding of the HLT Conference*.

H. Cui, R. Sun, K. Li, M.Y. Kan, and T.S. Chua. 2005. Question answering passage retrieval using dependency relations. In *Proceedings of the ACM SIGIR Conference*.

H.T. Dang, D. Kelly, and J. Lin. 2007. Overview of the TREC 2007 question answering track. *In Proceedings of the Text REtrieval Conference, TREC*.

A. Echihabi and D. Marcu. 2003. A noisy-channel approach to question answering. In *Proceedings of the ACL Conference*.

C. Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press.

D. Ferrucci, E. Brown, J. Chu-Carroll, J. Fan, D. Gondek, A. Kalyanpur, A. Lally, J. Murdock, E. Nyberg, J. Prager, N. Schlaefer, and C. Welty. 2010. Building Watson: An overview of the DeepQA project. *AI Magazine*, 31(3):59–79.

J. R. Finkel, T. Grenager, and C. D. Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL*.

M.H. Heie, E.W.D. Whittaker, and S. Furui. 2011. Question answering using statistical language modelling. *Computer Speech & Language*.

P. Koehn, H. Hoang, A. Mayne, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Annual Meeting of the ACL, Demonstration Session*.

T. Kudo and Y. Matsumoto. 2003. Fast methods for kernelbased text analysis. In *Proceedings of ACL Conference*.

J.T. Lee, S.B. Kim, Y.I. Song, and H.C. Rim. 2008. Bridging lexical gaps between queries and questions on large online Q&A collections with compact translation models. In *Proceedings of the EMNLP Conference*.

X. Li and D. Roth. 2005. Learning question classifiers: The role of semantic information. *JNLE*.

C-Y. Lin and F. Och. 2004. Automatic Evaluation of Machine Translation Quality Using Longest Common Subsequence and Skip-Bigram Statics. In *Proceedings of the ACL Conference*.

D. Moldovan, C. Clark, S. Harabagiu, and D. Hodges. 2007. Cogex: A semantically and contextually enriched logic prover for question answering. *Journal of Applied Logic*, 5(1).

V. Murdock and W.B. Croft. 2005. Simple translation models for sentence retrieval in factoid question answering. In *Procedings of the ACM SIGIR Conference*.

F. Och and H. Ney. 2002. Discriminative Training and Maximum Entropy Models for Statistical Machine Translation. In *Proceedings of the ACL Conference*.

F. Och and H. Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*.

F. Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the ACL Conference*.

K. Papineni, S. Roukos, T. Ward, and W-J. Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the ACL Conference*.

A. Peñas, E. Hovy, P. Forner, Á. Rodrigo, R. Sutcliffe, C. Forascu, and C. Sporleder. 2011. Overview of QA4MRE at CLEF 2011: Question answering for machine reading evaluation. *Working Notes of CLEF*.

D. Ravichandran and E. Hovy. 2002. Learning surface text patterns for a question answering system. In *Proceedings of the ACL Conference*.

S. Riezler, A. Vasserman, I. Tsochantaridis, V. Mittal, and Y. Liu. 2007. Statistical machine translation for query expansion in answer retrieval. In *Proceedings of the ACL Conference*.

T. Sakai, N. Kando, C.J. Lin, T. Mitamura, H. Shima, D. Ji, K.H. Chen, and E. Nyberg. 2008. Overview of the NTCIR-7 ACLIA IR4QA task. In *Proceedings of NTCIR Conference*.

R. Soricut and E. Brill. 2004. Automatic question answering: Beyond the factoid. In *Proceedings of HLT-NAACL Conference*.

A. Stolcke. 2002. SRILM – An extensible language modeling toolkit. In *Proc. Intl. Conf. on Spoken Language Processing*.

M. Surdeanu, D. Dominguez-Sal, and P.R. Comas. 2006. Design and performance analysis of a factoid question answering system for spontaneous speech transcriptions. *Proceedings of the INTERSPEECH Conference*.

M. Surdeanu, M. Ciaramita, and H. Zaragoza. 2011. Learning to rank answers to non-factoid questions from web collections. *Computational Linguistics*.

E.M. Voorhees. 2002. Overview of the TREC 2002 Question Answering track. In *In Proceedings of the Text REtrieval Conference, TREC*.