Learning to Translate for Multilingual Question Answering

Ferhan Ture
Comcast Labs*
1110 Vermont Ave NW Ste 600
Washington, DC, 20005 USA
ferhan_ture@cable.comcast.com

Elizabeth Boschee
Raytheon BBN Technologies
10 Moulton St
Cambridge, MA, 02138 USA
eboschee@bbn.com

Abstract

In multilingual question answering, either the question needs to be translated into the document language, or vice versa. In addition to *direction*, there are multiple *methods* to perform the translation, four of which we explore in this paper: word-based, 10-best, context-based, and grammar-based. We build a feature for each combination of translation *direction* and *method*, and train a model that learns optimal feature weights. On a large forum dataset consisting of posts in English, Arabic, and Chinese, our novel *learn-to-translate* approach was more effective than a strong baseline ($p < 0.05$): translating all text into English, then training a classifier based only on English (original or translated) text.

1 Introduction

Question answering (QA) is a specific form of the information retrieval (IR) task, where the goal is to find relevant well-formed answers to a posed question. Most QA pipelines consist of three main stages: (a) preprocessing the question and collection, (b) retrieval of candidate answers in the collection, and (c) ranking answers with respect to their relevance to the question and return the top N answers. The types of questions can range from factoid (e.g., "What is the capital of France?") to causal (e.g., "Why are trees green?"), and opinion questions (e.g., "Should USA lower the drinking age?").

The most common approach to *multilingual QA* (MLQA) has been to translate all content into its most probable English translation via machine translation (MT) systems. This strong baseline, which we refer to as *one-best MT* (1MT), has been successful in prior work (Hartrumpf et al., 2009; Lin and Kuo, 2010; Shima and Mitamura, 2010). However, recent advances in cross-lingual IR (CLIR) show that one can do better by representing the translation space as a probability distribution (Ture and Lin, 2014). In addition, MT systems perform substantially worse with user-generated text, such as web forums (Van der Wees et al., 2015), which provide extra motivation to consider alternative translation approaches for higher recall. To our knowledge, it has yet to be shown whether these recent advancements in CLIR transfer to MLQA.

We introduce a novel answer ranking approach for MLQA (i.e., *Learning to Translate* or L2T), a model that learns the optimal translation of question and/or candidate answer, based on how well it discriminates between good and bad answers. We achieve this by introducing a set of features that encapsulate lexical and semantic similarities between a question and a candidate answer through various translation strategies (Section 3.1). The model then learns feature weights for each combination of translation *direction* and *method*, through a discriminative training process (Section 3.2). Once a model is trained, it can be used for MLQA, by sorting each candidate answer in the collection by model score. Instead of learning a single model to score candidate answers in any language, it might be meaningful to train a separate model that can learn to discriminate between good and bad answers in each language. This can let each model learn *feature weights custom to*

[*] This work was completed while author was an employee of Raytheon BBN Technologies.

the language, therefore allowing a more fine-grained ranking (Section 3.4). We call this alternative approach *Learning to Custom Translate* (L2CT).

Experiments on the DARPA Broad Operational Language Technologies (BOLT) IR task[1] confirm that L2T yields statistically significant improvements over a strong baseline ($p < 0.05$), in three out of four experiments. L2CT outperformed the baseline as well, but was not more effective than L2T.

2 Related Work

For the last decade or so, research in QA has mostly been driven by annual evaluation campaigns like TREC,[2] CLEF,[3] and NTCIR.[4] Most earlier work relied on either rule-based approaches where a set of rules were manually crafted for each type of question, or IR-like approaches where each pair of question and candidate answer was scored using retrieval functions (e.g., BM25 (Robertson et al., 2004)). On the other hand, training a classifier for ranking candidate answers allows the exploitation of various features extracted from the question, candidate answer, and surrounding context (Madnani et al., 2007; Zhang et al., 2007). In fact, an explicit comparison at 2007 TREC confirmed the superiority of machine learning-based (ML-based) approaches (F-measure 35.9% vs 38.7%) (Zhang et al., 2007). Learning-to-rank approaches have also been applied to QA successfully (Agarwal et al., 2012).

Previous ML-based approaches have introduced useful features from many aspects of natural language, including lexical (Brill et al., 2001; Attardi et al., 2001), syntactic (Alfonseca et al., 2001; Katz et al., 2005), semantic (Cui et al., 2005; Katz et al., 2005; Alfonseca et al., 2001; Hovy et al., 2001), and discourse features, such as coreference resolution (Morton, 1999), or identifying temporal/spatial references (Saquete et al., 2005; Harabagiu and Bejan, 2005), which are especially useful for "why" and "how" questions (Kolomiyets and Moens, 2011). Additionally, semantic role labeling and dependency trees are other forms of semantic analysis used widely in NLP applications (Shen and Lapata, 2007; Cui et al., 2005).

When dealing with multilingual collections, most prior approaches translate all text into English beforehand, then treat the task as monolingual retrieval (previously referred to as 1MT). At recent evaluation campaigns like CLEF and NTCIR,[5] almost all teams simply obtained the one-best question translation, treating some online MT system as a black box (Adafre and van Genabith, 2009; Hartrumpf et al., 2009; Martinez-Gonzalez et al., 2009; Lin and Kuo, 2010; Shima and Mitamura, 2010), with few notable exceptions that took term importance (Ren et al., 2010), or semantics (Munoz-Terol et al., 2009) into account. Even for non-factoid MLQA, most prior work does not focus on the translation component (Luo et al., 2013; Chaturvedi et al., 2014).

Contributions. Ture and Lin described three methods for translating queries into the collection language in a probabilistic manner, improving *document retrieval* effectiveness over a one-best translation approach (2014). Extending this idea to MLQA appears as a logical next step, yet most prior work relies solely on the one-best translation of questions or answers (Ko et al., 2010b; García-Cumbreras et al., 2012; Chaturvedi et al., 2014), or selects the best translation out of few options (Sacaleanu et al., 2008; Mitamura et al., 2006). Mehdad et al. reported improvements by including the top ten translations (instead of the single best) and computing a distance-based entailment score with each (2010). While Espla-Gomis et al. argue that using MT as a black box is more convenient (and modular) (2012), there are potential benefits from a closer integration between statistical MT and multilingual retrieval (Nie, 2010). To the best of our knowledge, there is no prior work in the literature, where the ***optimal query and/or answer translation is learned via machine learning***. This is our main contribution, with which we outperform the state of the art.

In addition to learning the optimal translation, we ***learn the optimal subset of the training data for a given task***, where the criteria of whether we include a certain data instance is based on either the source language of the sentence, or the language in which the sentence was annotated. Training data selection strategies have not been studied extensively in the

[1] http://www.darpa.mil/Our_Work/I2O/Programs

[2] http://trec.nist.gov

[3] http://www.clef-initiative.eu

[4] http://research.nii.ac.jp/ntcir/index.html

[5] Most recent MLQA tracks were in 2008 (CLEF) and 2010 (NTCIR).

QA literature, therefore the effectiveness of our simple language-related criteria can provide useful insights to the community.

When there are multiple independent approaches for ranking question-answer pairs, it is required to perform a *post-retrieval merge*: each approach generates a ranked list of answers, which are then merged into a final ranked list. This type of system combination approach has been applied to various settings in QA research. Merging at the document-level is common in IR systems (e.g., (Tsai et al., 2008)), and has shown to improve multilingual QA performance as well (García-Cumbreras et al., 2012). Many QA systems combine answers obtained by different variants of the underlying model (e.g., (Brill et al., 2001) for monolingual, (Ko et al., 2010a; Ko et al., 2010b) for multilingual QA). We are not aware, however, of any prior work that has explored the ***merging of answers that are generated by language-specific ranking models***. Although this does not show increased effectiveness in our experiments, we believe that it brings a new perspective to the problem.

3 Approach

Our work is focused on a specific stage of the QA pipeline, namely *answer ranking*: Given a natural-language question q and k candidate answers $s_1, \ldots, s_k$, we score each answer in terms of its relevance to q. In our case, candidate answers are sentences extracted from all documents retrieved in the previous stage of the pipeline (using Indri (Metzler and Croft, 2005)). Hereafter, sentence and answer might be used interchangeably.

While our approach is not language-specific, we assume (for simplicity) that questions are in English, whereas sentences are in either English, Arabic, or Chinese. Non-English answers are translated back into English before returning to user.

Our approach is not limited to any question type, factoid or non-factoid. Our main motivation is to provide good QA quality on any multilingual Web collection. This entails finding answers to questions where there is no single answer, and for which human agreement is low. We aim to build a system that can successfully retrieve relevant information from open-domain and informal-language content. In this scenario, two assumptions made by many of the prior approaches fail:

1) We can accurately classify questions via template patterns (Chaturvedi et al. argue that this does not hold for non-factoid questions (2014))

2) We can accurately determine the relevance of an answer, based on its automatic translation into English (Wees et al. show how recall decreases when translating user-generated text (2015))

To avoid these assumptions, we opted for a more adaptable approach, in which question-answer relevance is modeled as a function of features, intended to capture the relationship between the question and sentence text. Also, instead of relying solely on a single potentially incorrect English translation, we increase our chances of a hit by translating both the question and the candidate answer, using four different translation methods. Our main features, described throughout this section, are based on lexical similarity computed using these translations. The classifier is trained on a large number of question-answer pairs, each labeled by a human annotator with a binary relevance label.[6]

3.1 Representation

In MLQA, since questions and answers are in different languages, most approaches translate both into an intermediary language (usually English). Due to the error-prone nature of MT, valuable information often gets "lost in translation". These errors are especially noticeable when translating informal text or less-studied languages (Van der Wees et al., 2015).

Translation Direction. We perform a *two-way translation* to better retain the original meaning: in addition to translating each non-English sentence into English, we also translate the English questions into Arabic and Chinese (using multiple translation methods, described below). For each question-answer pair, we have two "views": comparing translated question to the original sentence (i.e., *collection-language* (CL) view); and comparing original question to the translated sentence (i.e., *question-language* (QL) view).

Translation Method. When translating text for retrieval tasks like QA, including a variety of alterna-

[6] Annotators score each answer from 1 to 5. We label any score of 3 or higher as relevant.

tive translations is as important as finding the most accurate translation, especially for non-factoid questions, where capturing (potentially multiple) underlying topics is essential. Recent work in cross-language IR (CLIR) has shown that incorporating probabilities from the internal representations of an MT system to "translate" the question can accomplish this, outperforming standard one-best translation (Ture and Lin, 2014). We hypothesize that these improvements transfer to multilingual QA as well.

In addition to *translation directions*, we explored four *translation methods* for converting the English question into a probabilistic representation (in Arabic and Chinese). Each method builds a probability distribution for every question word, expressing the translation space in the collection language. More details of first three methods can be found in (Ture and Lin, 2014), while fourth method is a novel query translation method adapted from the neural network translation model described in (Devlin et al., 2014).

Word: In MT, a word alignment is a many-to-many mapping between source- and target-language words, learned without supervision, at the beginning of the training pipeline (Och, 2003). These alignments can be converted into word translation probabilities for CLIR (Darwish and Oard, 2003).

For example, in an English-Arabic parallel corpus, if an English word appears m times in total and is aligned to a certain Arabic word k times, we assign a probability of $\frac{k}{m}$ for this translation. This simple idea has performed greatly in IR for generating a probability distribution for query word translations.

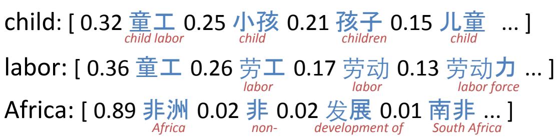
Grammar: Probabilities are derived from a synchronous context-free grammar, which is a typical translation model found in MT systems (Ture and Lin, 2014). The grammar contains rules r that follow the form $\alpha \mid \beta \mid \mathcal{A} \mid \ell(r)$, stating that source-language word α can be translated into target-language word β with an associated likelihood value $\ell(r)$ ($\mathcal{A}$ represents word alignments). For each rule r that applies to the question, we identify each source word s_j. From the word alignment information included in the rule, we can find all target words that s_j is aligned to. By processing all the rules to accumulate likelihood values, we construct translation probabilities for each word in the question.

10-best: Statistical MT systems retrieve a ranked list of translations, not a single best. Ture and Lin exploited this to obtain word translation probabilities from the top 10 translations of the question (2014). For each question word w, we can extract which grammar rules were used to produce the translation – once we have the rules, word alignments allow us to find all target-language words that w translates into. By doing this for each question translation, we construct a probability distribution that defines the translation space of each question word.

Context: Neural network-based MT models learn context-dependent word translation probabilities – the probability of a target word is dependent on the source word it aligns to, as well as a 5-word window of context (Devlin et al., 2014). For example, if the Spanish word "placer" is aligned to the English word "pleasure", the model will not only learn from this word-to-word alignment but also consider the source sentence context (e.g., "Fue un placer conocerte y tenerte unos meses."). However, since short questions might lack full sentence context, our model should have the flexibility to translate under partial or no context. Instead of training the NN-base translation model on full, well-formed sentences, we custom-fit it for question translation: words in the context window are randomly masked by replacing it with a special filler token <F>. This teaches the model how to accurately translate with full, partial context, or no context. For the above example, we generate partial contexts such as "fue un placer <F> y" or "<F> <F> placer conocerte y". Since there are many possibilities, if the context window is large, we randomly sample a few of the possibilities (e.g., 4 out of 9) per training word.

In Figure 1, we display the probabilistic structure produced the *grammar-based* translation method, when implemented as described above. Each English word in the question is translated into a probabilistic structure, consisting of Chinese words and corresponding probabilities that represent how much weight the method decides to put on that specific word. Similar structures are learned with the other three translation methods.

We are not aware of any other MLQA approach that represents the question-answer pair based on their probabilistic translation space.



child: [0.32 童工 0.25 小孩 0.21 孩子 0.15 儿童 ...]
labor: [0.36 童工 0.26 劳工 0.17 劳动 0.13 劳动力 ...]
Africa: [0.89 非洲 0.02 非 0.02 发展 0.01 南非 ...]

Figure 1: Probabilistic grammar-based translation of example question. The example question "Tell me about child labor in Africa" is simplified by our preprocessing engine to "child labor africa".

3.2 Features

Given two different translation directions (*CL* and *QL*), and four different translation methods (*Word*, *Grammar*, *10-best*, *Context*), our strategy is to leverage a machine learning process to determine how helpful each signal is with respect to the end task. For this, we introduced separate question-answer similarity features based on each combination of translation direction and method.

Collection-language Features. In order to compute a single real-valued vector to represent the question in the collection language (*LexCL*), we start with the probabilistic structure representing the question translation (e.g., Figure 1 is one such structure when the translation method is *grammar-based*). For each word in the collection-language vocabulary, we compute a weight by averaging its probability across the terms in the probabilistic structure.

$$v_{q_{\text{grammar}}}(w) = \text{avg}_i \text{Pr}(w|q_i) \qquad (1)$$

where w is a non-Engish word and $\text{Pr}(w|q_i)$ is the probability of w in the probability distribution corresponding to the i^{th} query term.

Figure 2 shows the real-valued vector computed based on the probabilistic question translation in Figure 1. The Chinese word translated as "child labor" has a weight of 0.32, 0.36, and 0 in the probability distributions of the three query terms, respectively. Averaging these three values results in the final weight of 0.23 in $v_{q_{\text{grammar}}}$ in Figure 2. Notice that these weights are normalized by construction.

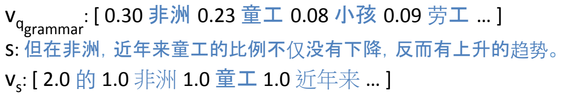
Similarly, a candidate answer s in Chinese is represented by normalized word frequencies:

$$v_s(w) = \frac{\text{freq}(w|s)}{\sum_{w'} \text{freq}(w'|s)} \qquad (2)$$

Feature category	**Question repr. ($v_{q'}$)**	**Sentence repr. ($v_{s'}$)**	**Feature Value**
LexCL	$v_{q_{\text{word}}}$	v_s	cosine $(v_{q'}, v_{s'})$
	$v_{q_{\text{10best}}}$	v_s	
	$v_{q_{\text{context}}}$	v_s	
	$v_{q_{\text{grammar}}}$	v_s	
LexQL	v_q	$v_{s_{\text{1best}}}$	

Table 1: List of features used in L2T, and how the values are computed from vector representations.

Given the two vectors, we compute the cosine similarity. Same process is repeated for the other three translation methods. The four lexical collection-language similarity features are collectively called *LexCL*.



$v_{q_{\text{grammar}}}$: [0.30 非洲 0.23 童工 0.08 小孩 0.09 劳工 ...]
s: 但在非洲，近年来童工的比例不仅没有下降，反而有上升的趋势。
v_s: [2.0 的 1.0 非洲 1.0 童工 1.0 近年来 ...]

Figure 2: Vector representation of grammar-translated question (q_{grammar}) and sentence (s).

Question-language Features. As mentioned before, we also obtain a similarity value by translating the sentence (s_{1best}) and computing the cosine similarity with the original question (q). v_q and $v_{s_{\text{1best}}}$ are computed using Equation 2. Although it is possible to translate the sentence into English using the same four methods, we only used the one-best translation due to the computational cost. Hence, we have only one lexical similarity feature in the QL view (call *LexQL*).

The computation process for the five *lexical similarity* features is summarized in Table 1. After computation, feature weights are learned via a maximum-entropy model.[7] Although not included in the figure or table, we also include the same set of features from the sentence preceding the answer (within the corresponding forum post), in order to represent the larger discourse.

3.3 Data Selection

In order to train a machine learning model with our novel features, we need positive and negative examples of question-answer pairs (i.e., (q, s)). For this,

[7] Support vector machines yielded worse results.

for each training question, our approach is to hire human annotators to label sentences retrieved from the non-English collections used in our evaluation. It is possible to label the sentences in the source language (i.e., Arabic or Chinese) or in the question language (i.e., translated into English). In this section, we explore the question of whether it is useful to distinguish between these two independently created labels, and whether this redundancy can be used to improve the machine learning process.

We hypothesize two reasons why selecting training data based on language might benefit MLQA:
i) The translation of non-English candidate answers might lack in quality, so annotators are likely to judge some relevant answers as non-relevant. Hence, training a classifier on this data might lead to a tendency to favor English answers.
ii) For the question-answer pairs that were annotated in both languages, we can remove noisy (i.e., labeled inconsistently by annotators) instances from the training set.

The question of annotation is an unavoidable part of evaluation of MLQA systems, so finding the optimal subset for training is a relevant problem. In order to explore further, we generated six subsets with respect to (a) the original *language* of the answer, or (b) the language of *annotation* (i.e., based on original text or its English translation):
en: Sentences from the English corpus.
ar/ch: Sentences from the Arabic / Chinese corpus (regardless of how it was judged).
consist: All sentences except those that were judged inconsistently.
src+: Sentences judged only in original text, or judged in both consistently.
en+: Sentences that are either judged only in English, or judged in both original and English translation consistently.
all: All sentences.

These subsets were determined based on linguistically motivated heuristics, but choosing the most suitable one (for a given task) is done via machine learning (see Section 4).

3.4 Language-specific Ranking

Scoring Arabic sentences with respect to a question is inherently different than scoring English (or Chinese) sentences. The quality of resources, grammar, etc., as well as other internal dynamics might differ greatly across languages. We hypothesize that there is no one-size-fits-all model, so the parameters that work best for English retrieval might not be as useful when scoring sentences in Arabic, and/or Chinese.

Our proposed solution is to apply a separate classifier, custom-tuned to each collection, and retrieve three *single-language ranked lists* (i.e., in English, Arabic, and Chinese). In addition to comparing each custom-tuned, language-specific classifier to a single, language-independent one, we also use this idea to propose an approach for MLQA:
`L2CT`(n) Retrieve answers from each language using separate classifiers (call these lists English-only, Arabic-only, and Chinese-only), take the best answers from each language, then merge them into a mixed-language set of n answers.

We compare this to the standard approach:
`L2T`(n) Retrieve up to n mixed-language answers using a single classifier.

Four heuristics were explored for merging lists in the `L2CT` approach.[8] Two common approaches are *uniform* and *alternate* merging (Savoy, 2004):
Uniform: A straightforward merge can be achieved by using the classifier scores (i.e., probability of answer relevance, given question) to sort all answers, across all languages, and include the top n in the final list of answers. Classifier scores are normalized into the [0,1] range for comparability.
Alternate: We alternate between the lists, picking one answer at a time from each, stopping when the limit n has been reached.

Since answers are expected in English, there is a natural preference for answers that were originally written English, avoiding noisy text due to translation errors. However, it is also important not to restrict answers entirely to English sources, since that would defeat the purpose of searching in a multilingual collection. We implemented the following methods to account for language preferences:
English first: We keep all sufficiently-confident (i.e., normalized score above a fixed threshold) answers from the English-only list first, and start including answers from Arabic- and Chinese-only lists only if

[8] In addition to these heuristics, the optimal merge could be learned from training data, as a "learning to rank" problem. This is out of the scope of this paper, but we plan to explore the idea in the future.

the limit of n answers has not been reached.
Weighted: Similar to *Uniform*, but we weight the normalized scores before sorting. The optimal weights can be learned by using a grid-search procedure and a cross-validation split.

4 Evaluation

In order to perform controlled experiments and gain more insights, we split our evaluation into four separate tasks: three tasks focus on retrieving answers from posts written in a specified language (*English-only*, *Arabic-only*, or *Chinese-only*) [9], and the last task is not restricted to any language (*Mixed-language*). All experiments were conducted on the DARPA BOLT-IR task. The collection consists of 12.6M Arabic, 7.5M Chinese, and 9.6M English Web forum posts. All runs use a set of 45 non-factoid (mostly opinion and causal) English questions, from a range of topics. All questions and forum posts were processed with an information extraction (IE) toolkit (Boschee et al., 2005), which performs sentence-splitting, named entity recognition, coreference resolution, parsing, and part-of-speech tagging.

All non-English posts were translated into English (one-best only), and all questions were translated into Arabic and Chinese (probabilistic translation methods from Section 3.1). For all experiments, we used the same state-of-the-art English↔Arabic (En-Ar) and English↔Chinese (En-Ch) MT systems (Devlin et al., 2014). Models were trained on parallel corpora from NIST OpenMT 2012, in addition to parallel forum data collected as part of the BOLT program (10M En-Ar words; 30M En-Ch words). Word alignments were learned with GIZA++ (Och and Ney, 2003) (five iterations of IBM Models 1–4 and HMM).

After all preprocessing, features were computed using the original post and question text, and their translations. Training data were created by having annotators label all sentences of the top 200 documents retrieved by Indri from each collection (for each question). Due to the nature of retrieval tasks, training data usually contains an unbalanced portion of negative examples. Hence, we split the data into balanced subsets (each sharing the same set of positively labeled data) and train multiple classifiers, then take a majority vote when predicting.

For testing, we froze the set of candidate answers and applied the trained classifier to each question-answer pair, generating a ranked list of answers for each question. This ranked list was evaluated by average precision (AP).[10] Due to the size and redundancy of the collections, we sometimes end up with over 1000 known relevant answers for a question. So it is neither reasonable nor meaningful to compute AP until we reach 100% recall (e.g., 11-point AP) for these cases. Instead, we computed AP-k, by accumulating precision values at every relevant answer until we get k relevant answers.[11] In order to provide a single metric for the test set, it is common to report the mean average precision (MAP), which in this case is the average of the AP-k values across all questions.

Baseline. As described earlier, the baseline system computes similarity between question text and the one-best translation of the candidate answer (we run the sentence through our state-of-the-art MT system). After translation, we compute similarity via scoring the match between the parse of the question text and the parse of the candidate answer, using our finely-tuned IE toolkit [reference removed for anonymization]. This results in three different similarity features: matching the tree node similarity, edge similarity, and full tree similarity. Feature weights are then learned by training this classifier discriminatively on the training data described above. This already performs competitively, outperforming the simpler baseline where we compute a single similarity score between question and translated text, and matching the performance of the system by Chaturvedi et al. on the BOLT evaluation (2014). Baseline MAP values are reported on the leftmost column of Table 2.

Data effect. In the baseline approach, we do not perform any data selection, and use all available data for training the classifier. In order to test our hypothesis that selecting a linguistically-motivated subset of the training data might help, we used 10-

[9] Shortened as *Eng*, *Arz*, and *Cmn*, respectively.

[10] Many other metrics (e.g., NDCG, R-precision) were explored during BOLT, and results were very similar.

[11] k was fixed to 20 in our evaluation, although we verified that conclusions do not change with varying k.

fold cross-validation to choose the optimal data set (among seven options described in Section 3.3). Results indicate that including English or Arabic sentences when training a classifier for Chinese-only QA is a bad idea, since effectiveness increases when restricted to Chinese sentences (`lang=ch`). On the other hand, for the remaining three tasks, the most effective training data set is `annot=en+consist`. These selections are consistent across all ten folds, and the difference is statistically significant for all but Arabic-only. The second column in Table 2 displays the MAP achieved when data selection is applied before training the baseline model.

Feature effect. To measure the impact of our novel features, we trained classifiers using either *LexCL*, *LexQL*, or *both* feature sets (Section 3.2). In these experiments, the data is fixed to the optimal subset found earlier. Results are summarized on right side of Table 2. Statistically significants improvements over *Baseline/Baseline+Data selection* are indicated with single/double underlining.

For Arabic-only QA, adding *LexQL* features yields greatest improvements over the baseline, while the same statement holds for *LexCL* features for the Chinese-only task. For the English-only and mixed-language tasks, the most significant increase in MAP is observed with all of our probabilistic bilingual features. For all but Arabic-only QA, the MAP is statistically significantly better ($p < 0.05$) than the baseline; for Chinese-only and mixed-language tasks, it also outperforms baseline plus data selection ($p < 0.05$).[12] All of this indicates the effectiveness of our probabilistic question translation, as well as our data selection strategy.

Task	Base	+Data	+Feats
Cmn	0.416	0.425 (*ch*)	0.451 (*LexCL*)
Arz	0.421	0.423 (*en+*)	0.425 (*LexQL*)
Eng	0.637	0.657 (*en+*)	0.660 (*all*)
Mixed	0.665	0.675 (*en+*)	0.681 (*all*)

Table 2: L2T evaluated using MAP with 10-fold cross-validation for each task. A statistically significant increase over Baseline/Base+Data is shown by single/double underlining ($p < 0.05$).

Understanding the contribution of each of the four *LexCL* features is also important. To gain insight, we trained a classifier using all *LexCL* features (using the optimal data subset learned earlier for each task), and then incrementally removed one of the features, and tested on the same task. This controlled experiment revealed that the *word* translation feature is most useful for Chinese-only QA (i.e., removing it produces largest drop in MAP, 0.6 points), whereas *context* translation appears to be most useful (by a slighter margin) in Arabic-only QA. In the former case, the diversity provided by word translation might be useful at increasing recall in retrieving Chinese answers. In retrieving Arabic answers, using context to disambiguate the translation might be useful at increasing precision. This result further emphasizes the importance of a customized translation approach for MLQA.

Furthermore, to test the effectiveness of the probabilistic translation approach (Section 3.1), we replaced all *LexCL* features with a single lexical similarity feature computed from the one-best question translation. This resulted in lower MAP: 0.427 to 0.423 for Arabic-only, and 0.451 to 0.425 for Chinese-only task ($p < 0.01$), supporting the hypothesis that *probabilistic translation is more effective than the widely-used one-best translation*. In fact, almost all gains in Chinese-only QA seems to be coming from the probabilistic translation.

For a robustness test, we let cross-validation select the best combination of (*data, feature*), mimicking a less controlled, real-world setting. In this case, the best MAP for the Arabic-, Chinese-, English-only, and Mixed-language tasks are 0.403, 0.448, 0.657, and 0.679, respectively. In all but Arabic-only, these are statistically significantly better ($p < 0.05$) than not tuning the feature set or training data (i.e., Baseline). This result suggests that our approach can be used for any MLQA task out of the box, and provide improvements.

Learning to Custom Translate (`L2CT`). We took the ranked list of answers output by each language-specific model, and merged all of them into a ranked list of mixed-language answers. For the *weighted* heuristic, we tried three values for the weight. In Table 3, we see that training separate classifiers for each subtask does not bring overall improve-

[12] Note that bilingual features are not expected to help on the English-only task, and the improvements come solely from data selection.

ments to the end task. Amongst merging strategies, the most effective were *weighted* (weights for each query learned by performing a grid-search on other queries) and *English first* – however, both are statistically indistinguishable from the single classifier baseline. In the latter case, the percentage of English answers is highest (88%), which might not be desirable. Depending on the application, the ratio of languages can be adjusted with an appropriate merging method. For instance, *alternate* and *norm* heuristics tend to represent languages almost equally.

Approach			**(En-Ch-Ar) %**	**MAP**
L2T			64-19-16	0.681
L2CT	Uniform		24-35-41	0.548
	Alt.		32-34-34	0.574
	Eng. First		88-6-6	0.668
	Weight	2	37-30-34	0.599
		5	51-24-25	0.654
		10	61-20-19	0.669

Table 3: L2T vs. L2CT for multilingual QA.

Even though we get lower MAP in the overall task, Table 2 suggests that it is worthwhile customizing classifiers for each subtask (e.g., the Chinese responses in the ranked list of L2CT are more relevant than *Single*.). The question of how to effectively combine the results into a mixed-language list, however, remains an open question.

5 Conclusions

We introduced L2T, a novel approach for MLQA, inspired from recent success in CLIR research. To our knowledge, this is the first use of probabilistic translation methods for this task, and the first attempt at using machine learning to learn the optimal question translation.

We also proposed L2CT, which uses language-specific classifiers to treat the ranking of English, Arabic, and Chinese answers as three separate subtasks, by applying a separate classifier for each language. While post-retrieval merging has been studied in the past, we have not come across any work that applies this idea specifically to create a language-aware ranking for MLQA.

Our experimental analysis shows the importance of data selection when dealing with annotations on source and translated text, and the effect of combining translation methods. L2T improved answer ranking effectiveness significantly for Chinese-only, English-only, and mixed-language QA.

Although results did not support the hypothesis that *learning a custom classifier* for the retrieval of each language would outperform the *single classifier* baseline, we think that more research is needed to fully understand how language-specific modeling can benefit MLQA. More sophisticated merging of multiple ranked lists of answers need to be explored. Learning to rank between answers from different languages might be more effective than heuristics. This would allow us to predict the final language ratio, based on many features (e.g., general collection statistics, quality of candidate answers, question category and complexity, MT system confidence levels) to merge question-answer pairs.

An even more comprehensive use of machine learning would be to learn word-level translation scores, instead of relying on translation probabilities from the bilingual dictionary, resulting in a fully customized translation. Similar approaches have appeared in learning-to-rank literature for monolingual IR (Bendersky et al., 2010), but not for multilingual retrieval. Another extension of this work would be to apply the same translation for translating answers into the question language (in addition to question translation). By doing this, we would be able to capture the semantics of each answer much better, since we have discussed that one-best translation discards a lot of potentially useful information.

Finally, since one of the take-away messages of our work is that a deeper understanding of linguistic context can improve QA effectiveness via more sophisticated question translation, we are hoping to see even more improvements by creating features based on word embeddings. One potential next step is to learn bilingual embeddings directly for the task of QA, for which we have started adapting some related work (Bai et al., 2010).

Acknowledgements

Jacob Devlin has provided great help in the design and implementation of the context-based question translation approach. We would also like to thank the anonymous reviewers for their helpful feedback.

References

SisayFissaha Adafre and Josef van Genabith. 2009. Dublin city university at qaclef 2008. In Carol Peters, Thomas Deselaers, Nicola Ferro, Julio Gonzalo, GarethJ.F. Jones, Mikko Kurimo, Thomas Mandl, Anselmo Peñas, and Vivien Petras, editors, *Evaluating Systems for Multilingual and Multimodal Information Access*, volume 5706 of *Lecture Notes in Computer Science*, pages 353–360. Springer Berlin Heidelberg.

Arvind Agarwal, Hema Raghavan, Karthik Subbian, Prem Melville, Richard D Lawrence, David C Gondek, and James Fan. 2012. Learning to Rank for Robust Question Answering. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 833–842, New York, NY, USA. ACM.

Enrique Alfonseca, Marco De Boni, José-Luis Jara-Valencia, and Suresh Manandhar. 2001. A prototype question answering system using syntactic and semantic information for answer retrieval. In *TREC*.

Giuseppe Attardi, Antonio Cisternino, Francesco Formica, Maria Simi, and Alessandro Tommasi. 2001. Piqasso: Pisa question answering system. In *TREC*.

Bing Bai, Jason Weston, David Grangier, Ronan Collobert, Kunihiko Sadamasa, Yanjun Qi, Olivier Chapelle, and Kilian Q. Weinberger. 2010. Learning to rank with (a lot of) word features. *Inf. Retr.*, 13(3):291–314.

Michael Bendersky, Donald Metzler, and W. Bruce Croft. 2010. Learning concept importance using a weighted dependence model. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, WSDM '10, pages 31–40, New York, NY, USA. ACM.

Elizabeth Boschee, Ralph Weischedel, and Alex Zamanian. 2005. Automatic information extraction. In *Proceedings of the International Conference on Intelligence Analysis*, volume 71.

Eric Brill, Jimmy Lin, Michele Banko, Susan Dumais, and Andrew Ng. 2001. Data-intensive question answering. In *In Proceedings of the Tenth Text REtrieval Conference (TREC*, pages 393–400.

Snigdha Chaturvedi, Vittorio Castelli, Radu Florian, Ramesh M Nallapati, and Hema Raghavan. 2014. Joint Question Clustering and Relevance Prediction for Open Domain Non-factoid Question Answering. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14, pages 503–514, New York, NY, USA. ACM.

Hang Cui, Renxu Sun, Keya Li, Min-Yen Kan, and Tat-Seng Chua. 2005. Question answering passage retrieval using dependency relations. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '05, pages 400–407, New York, NY, USA. ACM.

Kareem Darwish and Douglas W. Oard. 2003. Probabilistic structured query methods. In *SIGIR*.

Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard M. Schwartz, and John Makhoul. 2014. Fast and robust neural network joint models for statistical machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA*, pages 1370–1380.

Miquel Esplà-Gomis, Felipe Sánchez-Martínez, and Mikel L Forcada. 2012. UAlacant: Using Online Machine Translation for Cross-lingual Textual Entailment. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics - Volume 1: Proceedings of the Main Conference and the Shared Task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, SemEval '12, pages 472–476, Stroudsburg, PA, USA. Association for Computational Linguistics.

M Á García-Cumbreras, F Martínez-Santiago, and L A Ureña López. 2012. Architecture and Evaluation of BRUJA, a Multilingual Question Answering System. *Inf. Retr.*, 15(5):413–432, October.

Sanda Harabagiu and Cosmin Adrian Bejan. 2005. Question answering based on temporal inference. In *Proceedings of the AAAI-2005 workshop on inference for textual question answering*, pages 27–34.

Sven Hartrumpf, Ingo GlÃckner, and Johannes Leveling. 2009. Efficient question answering with question decomposition and multiple answer streams. In Carol Peters, Thomas Deselaers, Nicola Ferro, Julio Gonzalo, GarethJ.F. Jones, Mikko Kurimo, Thomas Mandl, Anselmo PeÃ±as, and Vivien Petras, editors, *Evaluating Systems for Multilingual and Multimodal Information Access*, volume 5706 of *Lecture Notes in Computer Science*, pages 421–428. Springer Berlin Heidelberg.

Eduard Hovy, Laurie Gerber, Ulf Hermjakob, Chin-Yew Lin, and Deepak Ravichandran. 2001. Toward semantics-based answer pinpointing. In *Proceedings of the first international conference on Human language technology research*, pages 1–7. Association for Computational Linguistics.

Boris Katz, Gary Borchardt, and Sue Felshin. 2005. Syntactic and semantic decomposition strategies for question answering from multiple resources. In *Proceedings of the AAAI 2005 workshop on inference for textual question answering*, pages 35–41.

Jeongwoo Ko, Luo Si, and Eric Nyberg. 2010a. Combining Evidence with a Probabilistic Framework for

Answer Ranking and Answer Merging in Question Answering. *Inf. Process. Manage.*, 46(5):541–554, September.

Jeongwoo Ko, Luo Si, Eric Nyberg, and Teruko Mitamura. 2010b. Probabilistic Models for Answer-ranking in Multilingual Question-answering. *ACM Trans. Inf. Syst.*, 28(3):16:1—-16:37, July.

Oleksandr Kolomiyets and Marie-Francine Moens. 2011. A survey on question answering technology from an information retrieval perspective. *Information Sciences*, 181(24):5412–5434.

Chuan-Jie Lin and Yu-Min Kuo. 2010. Description of the ntou complex qa system. In *Proceedings of NTCIR-8 Workshop*.

Xiaoqiang Luo, Hema Raghavan, Vittorio Castelli, Sameer Maskey, and Radu Florian. 2013. Finding what matters in questions. In *Proceedings of NAACL-HLT'13*.

N Madnani, Jimmy Lin, and Bonnie J Dorr. 2007. TREC 2007 ciQA Task: University of Maryland. *Proceedings of TREC*.

Ãngel Martinez-Gonzalez, Cesar de Pablo-Sanchez, Concepcion Polo-Bayo, MarÃaTeresa Vicente-Diez, Paloma Martinez-Fernandez, and Jose Luis Martinez-Fernandez. 2009. The miracle team at the clef 2008 multilingual question answering track. In Carol Peters, Thomas Deselaers, Nicola Ferro, Julio Gonzalo, GarethJ.F. Jones, Mikko Kurimo, Thomas Mandl, Anselmo PeÃ±as, and Vivien Petras, editors, *Evaluating Systems for Multilingual and Multimodal Information Access*, volume 5706 of *Lecture Notes in Computer Science*, pages 409–420. Springer Berlin Heidelberg.

Yashar Mehdad, Matteo Negri, and Marcello Federico. 2010. Towards Cross-lingual Textual Entailment. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 321–324, Stroudsburg, PA, USA. Association for Computational Linguistics.

Donald Metzler and W Bruce Croft. 2005. A Markov random field model for term dependencies. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '05, pages 472–479, New York, NY, USA. ACM.

Teruko Mitamura, Mengqiu Wang, Hideki Shima, and Frank Lin. 2006. Keyword translation accuracy and cross-lingual question answering in chinese and japanese. In *Proceedings of the Workshop on Multilingual Question Answering*, MLQA '06, pages 31–38, Stroudsburg, PA, USA. Association for Computational Linguistics.

Thomas S Morton. 1999. Using Coreference for Question Answering. In *Proceedings of the Workshop on Coreference and Its Applications*, CorefApp '99, pages 85–89, Stroudsburg, PA, USA. Association for Computational Linguistics.

Rafael Munoz-Terol, Marcel Puchol-Blasco, Maria Pardino, Jose Manuel Gomez, Sandra Roger, Katia Vila, Antonio Ferrandez, Jesus Peral, and Patricio Martinez-Barco. 2009. Integrating logic forms and anaphora resolution in the aliqan system. In *Evaluating Systems for Multilingual and Multimodal Information Access*, LNCS, pages 438–441. Springer Berlin Heidelberg.

Jian-Yun Nie. 2010. Cross-language information retrieval. *Synthesis Lectures on Human Language Technologies*, 3(1):1–125.

Franz J. Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 160–167, Stroudsburg, PA, USA. Association for Computational Linguistics.

Han Ren, Donghong Ji, and Jing Wan. 2010. Whu question answering system at ntcir-8 aclia task. In *Proceedings of NTCIR-8 Workshop*.

Stephen Robertson, Hugo Zaragoza, and Michael Taylor. 2004. Simple {BM25} extension to multiple weighted fields. In *Proc. CIKM*, pages 42–49.

Bogdan Sacaleanu, Günter Neumann, and Christian Spurk. 2008. Dfki-lt at qaclef 2008. In Carol Peters and et al., editors, *CLEF 2008 Working Notes*, Working Notes. Springer Verlag.

E Saquete, J L Vicedo, P Martínez-Barco, R Muñoz, and F Llopis. 2005. Evaluation of Complex Temporal Questions in CLEF-QA. In *Proceedings of the 5th Conference on Cross-Language Evaluation Forum: Multilingual Information Access for Text, Speech and Images*, CLEF'04, pages 591–596, Berlin, Heidelberg. Springer-Verlag.

Jacques Savoy. 2004. Combining multiple strategies for effective monolingual and cross-language retrieval. *Information Retrieval*, 7(1-2):121–148.

Dan Shen and Mirella Lapata. 2007. Using semantic roles to improve question answering. In *EMNLP-CoNLL*, pages 12–21. Citeseer.

Hideki Shima and Teruko Mitamura. 2010. Bootstrap pattern learning for open-domain clqa. In *Proceedings of NTCIR-8 Workshop*.

Ming-Feng Tsai, Yu-Ting Wang, and Hsin-Hsi Chen. 2008. A study of learning a merge model for multilingual information retrieval. In *Proceedings of the 31st*

Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '08, pages 195–202, New York, NY, USA. ACM.

Ferhan Ture and Jimmy Lin. 2014. Exploiting representations from statistical machine translation for cross-language information retrieval. *ACM Trans. Inf. Syst.*, 32(4):19:1–19:32, October.

Marlies Van der Wees, Arianna Bisazza, Wouter Weerkamp, and Christof Monz. 2015. What's in a domain? analyzing genre and topic differences in statistical machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 560–566, Beijing, China, July. Association for Computational Linguistics.

Chen Zhang, Matthew Gerber, Tyler Baldwin, Steven Emelander, Joyce Chai, and Rong Jin. 2007. Michigan State University at the 2007 TREC ciQA Task. In *Proceedings of the Sixteenth Text Retrieval Conference*, Gaithersburg, Maryland, November.