Alignment of Short Length Parallel Corpora with an Application to Web Search

Jitendra Ajmera*
IBM India Research Lab
New Delhi, India
jajmera1@in.ibm.com

Hema Swetha Koppula*
Cornell University
Ithaca, NY, USA
hema@cs.cornell.edu

Krishna P. Leela
Yahoo! Labs
Bangalore, India
krishna@yahoo-inc.com

Shibnath Mukherjee
Yahoo! Labs
Bangalore, India
shibm@yahoo-inc.com

Mehul Parsana
Yahoo! Labs
Bangalore, India
pmehul@yahoo-inc.com

ABSTRACT

With evolving Web, short length parallel corpora is becoming very common and some of these include user queries, web snippets etc. This paper concerns situations where short length parallel corpora has to be analyzed in order to find meaningful unit-alignment. This is similar to dealing with parallel corpora where a sentence level alignment of translations is required, but differs in that the alignment is to be inferred at unit (word or phrase) level. A Conditional Random Field (CRF) based approach is proposed to discover this unit alignment. Given pairs of semantically or syntactically similar entities, the problem is formulated as that of *mutual segmentation and sequence alignment problem*. The mutual segmentation refers to the process of segmenting the first entity based on units (or *labels*) in the second entity and vice-versa. The process of optimizing this mutual segmentation also results in optimal unit alignment. Since our training data is not segmented and unit-aligned, we modify the CRF objective function to accommodate unsupervised data and iterative learning. We have applied this framework to Web Search domain and specifically for query reformulation task. Finally, our experiments suggest that the proposed approach indeed results in meaningful alternatives of the original query.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Query formulation, Search process; I.7.0 [**Document and Text Processing**]: General

*The research described herein was conducted while the authors were at Yahoo! Labs.

CIKM'10, October 26–30, 2010, Toronto, Ontario, Canada.

General Terms

Algorithms, Performance

Keywords

Parallel corpora, Mutual segmentation, Sequence alignment, Web search, Query reformulation

1. INTRODUCTION

Analysis of parallel corpora often requires text alignment which identifies equivalent or similar text segments [3]. Traditionally parallel corpora is assumed to be text placed alongside its translation. In this case, text alignment identifies sentence-level equivalent or similar text segments.

This paper addresses a more general problem where the two halves of the parallel corpora are short text segments given to be semantically similar. These text segments may or may not be direct translations. The text alignment in this case would refer to identifying and mapping equivalent or similar "units". We refer to this as "mutual segmentation and sequence alignment" process.

Many application of such nature exist such as alignment of product titles across many pages in Information Extraction and alignment of units in user queries in Information Retrieval of Web search. In this paper, we consider Web search as our application domain with query reformulation as specific task. User queries tend to be short and may not express complete intent of the user. Even if it is complete, there may be other variants of the query which can result in even more meaningful search results. User queries, therefore, can be reformulated by replacing part of the query with another unit. Manually coming up with the list of these suitable replacements is a difficult task. It may be even more difficult if the data comes from another language. The solution presented in this paper can come up with this list of suitable replacements automatically.

The primary contribution of this paper is to define one single framework where segmentation and sequence alignment can be optimized iteratively. In contrast, if we consider the above-mentioned example of unit-mining for query reformulation, previous works have done away with the segmentation part of the problem by making some simplifying assumptions [5, 8].

In [5] only those query-pairs (parallel text) were consid-

ered which differed in one-unit. The differing units then became replacement candidates. Independence hypothesis likelihood ratio (LLR) was used to rank these unit-mappings. In [8], units were constrained to be words, primarily owing to the complexity of the algorithm. KL divergence of contextual models was used to find these unit-mappings in [8]. The proposed method is superior in that it does not only incorporate the segmentation, but also results in better alignment that optimizes a global objective function. Previous approaches as mentioned above essentially exploit normalized counts of two units occurring together in the parallel text optimizing only the local pair-wise mapping. Previous approaches also filter out a lot of useful information. For example, the information that the queries `britney spears mp3s` and `madonna songs` cannot be used for the mapping `mp3s<->songs` because there is no common unit.

In this paper, we formulate the problem as that of *mutual query segmentation and unit alignment*. Our approach starts with a dataset consisting of pairs of similar entities (parallel text) $\{(q, q')\}$. The *mutual segmentation* refers to the process of segmenting the entity q based on the content (units) of the entity q' and vice-versa. In the process of finding optimal mutual segmentation, we try to derive optimal unit-mappings and this step is referred to as *unit alignment*. In the example above, our mutual segmentation and alignment approach will try to find the mappings `{{britney spears<->madonna} {mp3s<->songs}}`.

Conditional random fields (CRFs) [6] have long been successfully used for both segmentation and alignment tasks for labeled data. Given the nature of the problem, CRF is considered with necessary modifications described below. We consider the units in q' as *labels* for the observation sequence q. The CRF training iteratively improves the segmentation of the two halves of the parallel text and discovers most relevant unit-mappings while optimizing a global objective function. Since our training data is not segmented and unit-aligned (labeled), we modify the CRF training process to accommodate unsupervised data and iterative learning.

After the CRF is trained, for every unit we have a sorted list of suggestions or *labels*. These unit-mappings are used for query reformulations. Given a query, we generate a list of *label sequences* by replacing every unit in the query by its suggestions. These sequences are sorted based on their conditional probabilities. To avoid any *concept-drift*, we restrict our approach to keeping only one unit-substitution in the label sequence and keep rest of the units in the original query unchanged.

In previous labeling and segmentation tasks using CRF, the target label set $\mathcal{L}$ is well defined and limited. Given an input observation sequence X, a number of different label sequences $\{Y|Y \in \mathcal{Y}\}$ can be generated by assigning different labels to different elements of X. CRF defines a conditional probability distribution $p(Y|X)$ and the label sequence maximizing this probability is assigned to the observation sequence X. CRF training generally takes a labeled or supervised training set $\{(X_i, Y_i)\}$ and estimates the parameters such that the total log likelihood of the data i.e. $\sum_i \log p(Y_i|X_i)$ is maximized.

Our formulation differs from the above approach in two ways. First, our label set $\mathcal{L}$ lies in the same space as the elements of the observation sequence and is much larger than what is considered in previous published works. This is because given parallel text (q, q'), we consider units in q' as labels for segmenting the observation sequence q and vice-versa. The second difference comes from the fact that we do not have labeled training data. Very few previous works on CRF have dealt with unsupervised training. An unsupervised text segmentation problem was solved using CRFs in [7]. However, structured reference tables are assumed provided. These tables are exploited to generate the CRF model for each attribute in the reference table.

The rest of the paper is organized as follows. Section 2 presents proposed segmentation and alignment using CRF. We present experimental evaluation in Section 3 and conclude in Section 4.

2. SEGMENTATION AND ALIGNMENT TECHNIQUE

Consider a parallel corpora where each element, $\{(q_l, q_r)\}$, is a pair of similar entities. Each entity comprises of some basic units. The segmentation of the entities in terms of these units is not available.

If we can somehow segment these entities in terms of their units, at least one of the units (potentially more) in q_l (u_{li}) should have mapping with unit(s) in q_r in order for the entities to be similar. Identifying these mappings is referred here as *sequence alignment* problem. It is easy to see that better unit-mappings will result in better segmentation and vice-versa.

Therefore, the segmentation and sequence-alignment steps can be repeated iteratively to result in optimal segmentation as well as optimal unit-mappings. In this paper, we propose a CRF based solution for this problem.

For a given pair (q_l, q_r), we start by considering units in q_r, (u_{rj}), as possible *labels* for the units in q_l, (u_{li}). Thus, we can hypothesize a set of label sequences $(\{Y \in q_r\})$ for q_l, such that each label sequence Y comprises of units (u_{rj}) in q_r . Let $f_k(q_l, Y, i)$ denote an indicator binary function which is equal to 1 if the i^{th} unit in q_l is equal to u_k and its label in Y is equal to u'_k, and equal to 0 otherwise. Let λ_k denote degree or importance of this indicator function. Thus, the number of parameters (λ_k) is equal to the number of unit pairs $\{(u_k, u'_k)\}$ in the training data. The conditional probability of a label sequence $Y \in \mathcal{Y}$ ($\mathcal{Y}$ denoting the set of all possible label sequences of q_l) is given by:

$$P(Y|q_l) = \frac{exp(\sum_i \sum_k \lambda_k f_k(q_l, Y, i))}{Z(q_l)} \quad (1)$$

The summation over i is basically summing over various units (u_{li}) in q_l. $Z(q_l)$ is the normalization constant (partition function) to make sure that:

$$\sum_{Y \in \mathcal{Y}} P(Y|q_l) = 1 \quad (2)$$

Consider the following example query pair that is given to be semantically similar:

`{five star hotels placeX,luxury accommodation cityY}`

In this case, the observation sequence q_l = `{five star hotels placeX}` can be segmented in various different ways such as:
seg1: `{five}{star hotels}{placeX}`
seg2: `{five star}{hotels}{placeX}`

seg3: `{five star}{hotels placeX}`
seg4: `{five placeX}{star}{hotels}`
....

Similarly, q_r = `{luxury accommodation cityY}` can be segmented to produce a number of different label sequences. Our goal is to train the CRF in such a way that when we compute:

$$Y^* = \underset{Y \in q_r}{\operatorname{argmax}} P(Y|q_l) \tag{3}$$

It should correspond to the mutual segmentation and alignment:

`{five star,luxury}{hotels,accommodation}{placeX,cityY}`

Note that label sequences are denoted by $Y \in q_r$ and are composed of units in q_r.

The process starts with considering all possible mutual segmentations of all the pairs $\{(q_l, q_r)\}$ in the corpora. This provides an exhaustive set of *labels* for each possible unit in the database. An indicator function $f_k(.)$ is hypothesized for each of these mappings and corresponding *weights* (λ_k) become parameters of the system that need to be estimated such that the following *total expected log likelihood* of the data is maximized.

$$L = \sum_{q_l} \sum_{Y \in q_r} \hat{P}(Y|q_l) \log P(Y|q_l) \tag{4}$$

where $\hat{P}(.)$ denotes the probability computation using current set of parameters such that:

$$\sum_{Y \in q_r} \hat{P}(Y|q_l) = 1 \tag{5}$$

Since we are dealing with unsupervised data (without segmentation and labels), our goal is to improve our estimates of $P(Y|q_l)$ (over $Y \in q_r$) and L (Eq. 4) iteratively.

We must also consider another important point about the variable length of the resulting segmentation and therefore the label sequence here. In our solution, different label sequences Y may have different lengths. A sequence having higher number of labels would therefore have a tendency to result in higher probability (Eq. 1) compared to a shorter label sequence. To account for this, we introduce length-normalization in the probability computation as follows:

$$p(Y|q_l) = \frac{exp(\frac{\sum_i \sum_k \lambda_k f_k(q_l, Y, i)}{f\#(Y, q_l)})}{Z(q_l)} \tag{6}$$

where,

$$f\#(Y, q_l) = \sum_i \sum_k f_k(q_l, Y, i)$$

The total expected log likelihood (Eq. 4) as a function of the parameters (Λ) is now computed as:

$$L(\Lambda) = \sum_{q_l} \sum_{Y \in q_r} \hat{P}(Y|q_l) \log P(Y|q_l) \tag{7}$$

$$= \sum_{q_l} \sum_{Y \in q_r} \hat{P}(Y|q_l) \sum_i \sum_k \frac{\lambda_k f_k(q_l, Y, i)}{f\#(q_l, Y)} - \sum_{q_l} \log Z(q_l)$$

Solving for all the parameters $\{\lambda_k | \lambda_k \in \Lambda\}$ simultaneously in Eq. 7 becomes intractable. We follow the iterative scaling algorithm proposed in [6]. The length-normalization incorporated in the likelihood computation (Eq. 6) results in a simpler solution as follows:

$$\overbrace{\sum_{q_l} \sum_{Y \in q_r} \hat{P}(Y|q_l) \sum_i \sum_k \frac{f_k(q_l, Y, i)}{f\#(q_l, Y)}}^{E(f_k)} = \tag{8}$$

$$\underbrace{\sum_{q_l} \sum_{Y \in \mathcal{Y}} P(Y|q_l) \sum_i \sum_k \frac{f_k(q_l, Y, i)}{f\#(q_l, Y)} exp(\delta_k)}_{\tilde{E}(f_k)}$$

where δ_k is the update for parameter λ_k required to result in the increase in the lower-bound of the total expected log likelihood of the data.

In Eq. 8 $P(Y|q_l)$ is the probability estimate using current set of CRF parameters Λ. Note that this is different from $\hat{P}(Y|q_l)$ for two reasons: 1) $\sum_{Y \in q_r} \hat{P}(Y|q_l) = 1$ while $\sum_{Y \in \mathcal{Y}} P(Y|q_l) = 1$, and 2) The estimate of $P(Y|q_l)$ change with every iterative scaling step whereas the estimate of $\hat{P}(Y|q_l)$ would only change in the expectation step of the expectation-maximization (EM) iteration. Thus, every expectation step is followed by maximization step consisting of several sub-iterations of iterative scaling. The updated parameter(s) are then given by:

$$\lambda_k^{'} = \lambda_k + \delta_k \tag{9}$$

3. EXPERIMENTAL EVALUATION

As we considered *Web Search* as an application, in this section we provide experimental evaluation of the proposed approach utilizing the logs of Yahoo! Search [1]. In specific, we evaluate our approach for query reformulation task of web search and compare with unit substitutions approach of Jones et al. [5], referred as "LLR" Method in the rest of the section.

For generating unit substitutions, we used one week of Yahoo! Search logs. We use successive queries issued by a user as parallel text for query reformulation task. Parallel corpora contained about 2.3 Billion unique query pairs. From these we filtered those query pairs which have occurred together only once in order to remove noise. This resulted in 78.6 million query pairs. On the filtered data we used stratified sampling to randomly select 100K query pairs from each decile in terms of query pair frequency, giving us 1 million query pairs for training the CRF model. Using these unit suggestions, rewrites are generated for the queries in the evaluation set where the queries selected for evaluation is from a different day.

For the LLR Method, we used the filtered query pair set for generating the unit substitutions as described by Jones et al. [5]. As expected, the LLR method required more number of query pairs to generate the same number of unit substitutions as that of our method. This is because the LLR method imposes further constraints on the overlap between the two queries in a pair. The queries are segmented into units and the unit substitutions are generated from the query pairs where only one segment has changed. For generating query reformulations, the queries from the evaluation set are first segmented and then the units are replaced by

its substitutions. As in the CRF method, we allow only one unit to be replaced from the original query for generating the reformulations. These reformulations are ranked using the linear regression model specified in [5].

Editorial Judgment	CRF Method	LLR Method
Precise Match	23.51%	23.55%
Approximate Match	42.99%	35.75%
Possible Match	17.99%	24.30%
Clear Mismatch	15.51%	16.40%

Table 1: Distribution of editorial judgments

We compute Discounted Cumulative Gain (DCG) [4] to evaluate the quality of proposed technique for query reformulation. The query and reformulation pairs were judged by human annotators on a 4-point scale - *Precise Match, Approximate Match, Possible Match and Clear Mismatch*, as per the guidelines described in [5]. *Precise Match* and *Approximate Match* rewrites together is considered as Specific Rewriting which can be used to retrieve highly relevant results as the rewrites have very close meaning to the original query. As the rewrites with *Possible Match* label have some categorical relationship with the original query and hence preserve the user interests, the rewrites with label *Possible Match* along with Precise and Approximate Match is considered for Broad Rewriting. We randomly sampled a set of 1000 queries from our evaluation query set for which our and comparison methods have at least five suggestions. These queries along with top five suggestions were editorially labeled. Table 1 shows the distribution of labels for CRF and LLR. It can be seen that our method has higher percent of rewrites for both Specific Rewriting and Broad Rewriting tasks. CRF method generates 12.14% higher number of Specific Rewrites and 1.06% higher Broad Rewrites compared to the LLR method.

	CRF Method	LLR Method	% improvement on LLR Method
DCG@1	7.13	6.26	13.89
DCG@2	11.92	10.48	13.74
DCG@3	15.76	13.85	13.79
DCG@4	19.05	17.19	10.82
DCG@5	22.06	20.22	9.09

Table 2: Comparison of DCG values of CRF and LLR Method

The DCG for a query is defined as:

$$DCG@K(Q) = \sum_{i=1}^{K} \frac{g(i)}{\log(1+i)}$$

where $g(i)$ is the gain associated with labeling of the result at rank i and K is the maximum depth of results to be considered. This takes into account the importance of ordering by discounting the gain at higher ranks. The $DCG@K$ for a query set, also called as the mean $DCG@K$ value, is obtained by taking the arithmetic mean of the per-query $DCG@K$ values. Since we have 5 rewrites per query we calculate DCG for 5 ranks to evaluate all the methods. We use the gain values of 10,7,3 and 0 for the labels *Precise Match*, *Approximate Match*, *Possible Match* and *Clear Mismatch* respectively, following [2]. The DCG results from Table 2 shows that our method significantly improves the $DCG@K$ for all the five rewrites with a consistent improvement over 13.79% for the first 3 ranks compared to the LLR method.

Results from human evaluation concludes that the proposed approach is significantly better than LLR method.

4. CONCLUSIONS

In this paper we presented an approach for segmentation and alignment of short parallel text where one unit can be replaced by its appropriate unit. The unit-suggestions are generated from pairs of parallel text. We considered web search as an application to illustrate our approach where two queries in a user session are considered as parallel text.

We iteratively segment the queries based on their mutual content and derive meaningful unit-mappings from these improved segmentations. We have adapted the CRF framework to achieve this where both the segmentation and the unit-mapping (*labeling*) steps can be iterated to optimize a global objective function. We have modified the CRF training to accommodate unsupervised training and iterative learning.

5. REFERENCES

[1] Yahoo! search. `http://search.yahoo.com/`.

[2] A. Broder, P. Ciccolo, E. Gabrilovich, V. Josifovski, D. Metzler, L. Riedel, and J. Yuan. Online expansion of rare queries for sponsored search. In *WWW '09: Proceedings of the 18th international conference on World wide web*, pages 511–520, New York, NY, USA, 2009. ACM.

[3] J. Chen and J. yun Nie. Parallel web text mining for cross-language IR. In *In In Proc. of RIAO*, pages 62–77, 2000.

[4] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, 2002.

[5] R. Jones, B. Rey, O. Madani, and W. Greiner. Generating query substitutions. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 387–396, New York, NY, USA, 2006. ACM.

[6] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML*, pages 282–289, 2001.

[7] X. Li, Y.-Y. Wang, and A. Acero. Extracting structured information from user queries with semi-supervised conditional random fields. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 572–579, New York, NY, USA, 2009. ACM.

[8] X. Wang and C. Zhai. Mining term association patterns from search logs for effective query reformulation. In *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management*, pages 479–488, New York, NY, USA, 2008. ACM.