Quasi-Synchronous Dependence Model for Information Retrieval

Jae-Hyun Park, W. Bruce Croft and David A. Smith
Center for Intelligent Information Retrieval
Department of Computer Science
University of Massachusetts, Amherst, MA 01003
{jhpark,croft,dasmith}@cs.umass.edu

ABSTRACT

Incorporating syntactic features in a retrieval model has had very limited success in the past, with the exception of term dependencies. This paper presents a new term dependency modeling approach based on a dependency parsing technique used for both queries and documents. Our model is inspired by a quasi-synchronous stochastic process for machine translation [21]. It describes four different types of syntactic relationships between dependent terms and allows inexact matching between documents and queries to deal with possible syntactic transformations. We also propose a machine learning technique for predicting optimal parameter settings for a retrieval model incorporating the syntactic relationships. The results on TREC collections show that the quasi-synchronous dependence model can improve retrieval performance and outperform a strong state-of-art baseline when we use predicted optimal parameters.

Categories and Subject Descriptors

H.3.3 [**Information System**]: Information Search and Retrieval

General Terms

Algorithm, Experimentation, Performance

Keywords

Term Dependency, Quasi-synchronous Grammar, Weighting Retrieval Model

1. INTRODUCTION

Term dependency has been studied for several decades to improve the effectiveness of information retrieval. Although independence assumptions simplify retrieval models, terms are actually dependent upon each other within documents and within queries. Terms are used together to make more specific meanings or, sometimes, totally different meanings. Despite this, it has been difficult to develop retrieval models incorporating term dependence that show consistent improvements over models that assume term independence independence assumptions. One of the challenging issues in modeling term dependence is that a concept can be expressed in different ways syntactically as well as lexically. In describing a concept in their queries, people use different vocabularies than authors use in describing the same concepts in their documents [27]. Query term expansion techniques have been studied to deal with a lexical mismatch between queries and relevant documents.

Similarly, even if the vocabularies in queries and documents are identical to each other, their relationships can differ in various ways: specifically, in order, proximity, and grammatical relation. In order to take account of the variability in relationships of terms between queries and documents, successful term dependence models in previous work allowed inexact matches in the order and proximity of dependent terms [16, 23]. Figure 1 shows example sentences containing the words "chemical" and "weapons". Even though the concept implied by "chemical weapons" is similar, their orders, distances, and syntactic relations vary between the example sentences. If we strictly regulate the order, distance, and syntactic relation of dependent terms, we will miss "chemical weapons" in relevant documents. The sequential dependence model (SDM) can match these important dependent terms by allowing differences in their order and proximity [16].

However, current models typically ignore term dependence based on syntactic relationships in queries and often restrict

> Al-Rabta ***chemical weapons*** plant was uncovered and destroyed in a fire.
>
> ... mentions ricin, sarin, soman, or anthrax as a toxic ***chemical*** used as a ***weapon***.
>
> ... its ***chemical*** and biological ***weapons*** and nuclear program.
>
> He intends to produce not only ***chemical*** but also bacteriological ***weapons***.

Figure 1: Example sentences which shows syntactic variations in the term dependency of the concept "chemical weapons"

dependency relations to adjacent term pairs. If the example sentences in Figure 1 are used in a query, all dependencies except the first will be ignored because they are not adjacent. Gao et al. [7] propose a dependence language approach which tries to extract long-distance term dependencies by incorporating dependency structure information. Song et al. [22] and Lee et al. [12] also propose a model based on linguistic parsing By choosing syntactically related term pairs in a query as dependent terms, these models attempt to overcome the limitation of sequential dependencies. However, the models are still limited to a head-modifier relation in which two dependent terms are directly linked in a dependency structure. Thus, they would still ignore all dependencies in Figure 1 except the first.

In this paper, we propose a term dependence model inspired by a quasi-synchronous stochastic process developed by Smith and Eisner [21]. Synchronous grammars were proposed for machine translation to generate translated expressions or identify translation examples by aligning a parse tree in a source language to a parse tree in a target language [20]. Because of inherent incompatibility between a source language and target language, syntactic and lexical variations occur during translating from a source sentence to a target sentence. Thus, a synchronous model should be able to align a translation unit in a source language to different forms than that of the original [8]. Smith and Eisner [21] model several different types of syntactic configurations in a target language. A synchronous model allows parent-child words in a source language tree to be associated with words having different syntactic configurations in a target language tree.

This inexact matching process of the quasi-synchronous model has also shown significant improvements for other research tasks such as open domain question-answering(QA) [26] and paraphrasing [6]. The processes of selecting answer sentences and paraphrased sentences are interpreted as a free translation process between sentences in the same language. In a similar way, we adopt the quasi-synchronous stochastic approach and generalize it for information retrieval, where a target sentence (a query) is generated from a set of sentences (a document) instead of one sentence. By using the quasi-synchronous stochastic approach, we represent term dependence more flexibly and incorporate transformations of dependence relationships.

The remainder of this paper is organized as follows. Section 2 introduces previous work about term dependence models, syntactic features, and parameter optimization methods for integrating different retrieval models. In Section 3 we explain how our model covers syntactic variations between queries and documents. Next, Section 4 describes a method to predict optimal weights of models for a given query. In Section 5 we evaluate our proposed model. Finally, in Section 6 we conclude and outline future work.

2. RELATED WORK

Term dependency has long been studied in the field of information retrieval and has been based on various features such as co-occurrences of terms, proximities between terms, etc. [5, 7, 12, 16, 23, 25]. Dependence models based on syntactic parsing results have been proposed to capture long-distance term dependencies. Gao et al. [7] proposed a dependence language model in which term dependencies were selected based upon the linkage structure L of queries and documents. The fundamental idea of this model is that queries and documents are represented in the form of the hidden variable, an acyclic, planar, undirected linkage graph L as shown Figure 2. The dependence language model generates not only a query but also the linkages of the query as follows

Figure 2: The sample linkage graph L of a query in [7]. They consider only a direct linkage between a word and its head word.

$$P(Q|D) = \sum_{L} P(Q, L|D) = \sum_{L} P(L|D)P(Q|L, D) \quad (1)$$

Then, dependent terms are defined by the edges in a linkage graph L and $P(Q|L, D)$ is decomposed in the similar way with the bigram model.

$$P(Q|L, D) = P(q_h|D) \prod_{(i,j)\in L} P(q_j|q_i, L, D) \quad (2)$$

in which, q_h is the sentential head word of a query. q_i is the head word of q_j in L.

Maisonnasse et al. [13] extend this dependence language model using a syntactic and semantic analysis model. Lee et al. [12] also suggested a language model which is based on dependency parse trees generated by a linguistic parser. These models aim to solve the limitation of the language model approach in which models fail to detect long-distance dependencies when just replacing the unigram language model with a bigram or biterm language model. However, they still have the limitation that term dependencies are derived only from head-dependent term pairs or directly connected nodes in the linkage structure L. As shown Figure 1, a head-dependent relation of terms can be expressed using indirect relations without changing its meaning.

Song et al. [22] introduced *variability*, which represents the probability that a head-modifier term pair in a query will not have a head-modifier relation in a document. They observe that some head-modifier term pairs in queries have stronger relations with each other than other head-modifier term pairs. If a strongly tied term pair, e.g. "mutual" → "fund", is unseen in a document, it is unlikely to expect the strongly tied term pair be used in a different way. On the other hand, although a weakly tied term pair, e.g. "textile" → "product" is unseen in a document, we can expect that the term pair is used with a transformed relation in the document. Based on this assumption, they use predicted variabilities for an interpolation weight as follows.

$$\begin{aligned} P(w_i \to w_j|w_i, D) =& (1 - v_i) \cdot P(w_i \to w_j|w_i, D) + \\ & v_i \cdot P(w_i \to w_j|w_i, C) \end{aligned} \quad (3)$$

in which, v_i is the variability of a head-modifier q_i and q_j. Although the variability implies the possibility that there are the transformations of the syntactic relationships of dependent terms between queries and documents, the variability still does not consider the transformed term pairs explicitly.

Query 625: Gather any information that mentions ricin, sarin, soman, or anthrax as a toxic chemical used as a weapon.

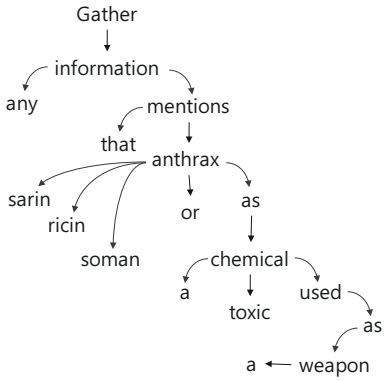


Figure 3: Example of syntactic parsing results (The description query of the TREC topic 625)

To identify dependencies of terms beyond head-modifier in both queries and documents and also to handle variations in dependence relations of terms, we adopt a relaxed alignment approach, specifically a quasi-synchronous grammar developed by Smith and Eisner [21]. As in Berger and Lafferty [3], we view the probability $p(Q|D)$ as the document-to-query mapping. We bring a quasi-synchronous approach to information retrieval and generalize it to adapt it for an alignment between a query and a document. In the next section, we briefly describe dependency parsing technique and then explain the quasi-synchronous model for information retrieval.

3. QUASI-SYNCHRONOUS MODEL

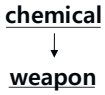
Dependency grammar is one of the ways to describe linguistic structure in which a grammar works directly in terms of dependencies between words themselves [14]. We need a parsing results to decide whether given two terms have a syntactically important relationship or not. The superstructure of a sentence such as phrasal or clausal structures are not our primary concern. Therefore, a dependency parsing approach focusing directly on words themselves is the most appropriate to satisfy our purpose. Dependency parsers output directed trees: each word in the sentence is has exactly one incoming edge, which comes from its 'head' or 'parent', with the except of the 'root' word, which has no incoming edges. These dependencies can be expressed as edges from a head word to its dependent word. Dependency parsing results can be depicted as a tree having the head word of a sentence as a root node (Figure 3). Smith and Eisner proposed a quasi-synchronous stochastic process [21] to allow parent and child words in a source language tree to be associated with words having different syntactic relations in a target language tree.

3.1 Quasi-Stochastic Process

An "synchronous" parsing model was developed to generate the parsing tree of a target sentence by matching the target sentence with a source sentence for a machine translation [20]. A synchronous model produces the parsing tree of a target sentence by recursively matching the child words

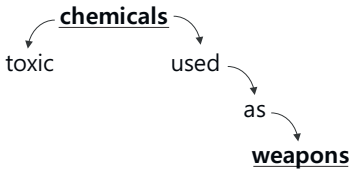
(a) *parent-child*

The inspectorate searched **chemical weapons.**



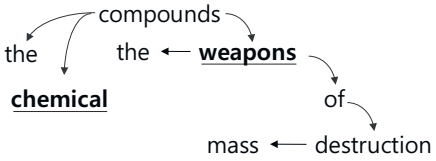
(b) *ancester-descendent*

The inspectorate searched toxic **chemicals** which is used as **weapons**.



(c) *siblings*

The inspectorate searched the **chemical** compounds, the **weapons** of mass destruction.



(d) *c-commanding*

The inspectorate searched the **chemical** compounds which is used as **weapons**.

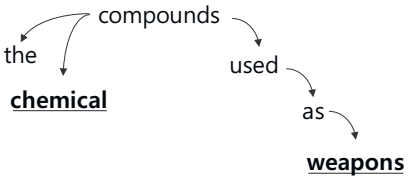


Figure 4: Four types of syntactic dependency configurations for the quasi-synchronous model. The quasi-synchronous model matches terms in queries and documents along with transformation between these dependence relations: (a) parent-child, (b) ascendant-descendant, (c) siblings, and (d) c-commanding.

of a given parent word in a corresponding source sentence. Words in a source language are not always translated into a target language in the same syntactic structure. Some source words may be translated into one target word and one source word may be translated into more than one word or a phrase. To solve these disagreements in source and target languages, the quasi-stochastic process allows words in a target sentence, which are aligned with a parent and child words in a source sentence, to have other relationships. In this paper, we consider the four configurations as follows:

- Parent-Child
- Ancestor-Descendent
- Siblings
- C-Commanding

Figure 4 shows examples for terms, "chemical" and "weapon".

Like the language model framework, the basic idea of the quasi-synchronous model is to rank a document using the probability that a query is generated by the document model. However, we infer a document model from the parsing results T_D of a document itself rather than the raw document D as in the dependence language model [7]. The document model generates not an individual term or a dependent term pair but a fragment of the parsing tree T_Q of a query Q through the loose alignment A.

$$P(Q|D) \approx P(T_Q, A|T_D) = P(A|T_D) \cdot P(T_Q|A, T_D) \quad (4)$$

where T_Q and T_D are the parsing trees of a query and a document, respectively. A is a set of possible combinations of the four syntactic configurations between a query and a document.

In previous work [7, 12, 22], the alignment A consists of only a parent-child relation. On the other hand, a quasi-synchronous approach allows inexact matching from all the four syntactic configurations to all. More specifically, when we consider the transformation of the syntactic configurations, being aligned with a parent-child (t_i, t_j) is different from being aligned with a child-parent (t_i, t_j). A parent-child, ancestor-descendent and c-commanding relations are different according to the order of two terms. Therefore, a dependent term pair in a query can be aligned to not four syntactic configurations but seven syntactic configurations and A has 4×7 elements for the combinations of syntactic configurations in a query and a document.

$$P(A|T_D)P(T_Q|A, T_D) = \sum_{(syn_D, syn_Q) \in A} P(syn_D, syn_Q|T_{D,syn_D}) \cdot P(T_{Q,syn_Q}|T_{D,syn_D}) \quad (5)$$

where syn_Q and syn_D are one of the four syntactic configurations and $P(syn_D, syn_Q|T_{D,syn_D})$ is the probability that a syntactic relation syn_D in a document is used in the form of syn_Q in a query. Intuitively, the probability that syn_D is transformed to syn_Q is different according to different terms as pointed out by [22]. For example, two words for a person's name are less likely to be used in the forms of ancestor-descendant, siblings, or c-commanding. However, to estimate the probabilities for a quasi-alignment, the number of parameters becomes impractically large. Therefore, for simplicity, we use a uniform distribution for $P(syn_D, syn_Q|T_{D,syn_D})$.

$$\sum_{(syn_D, syn_Q) \in A} P(syn_D, syn_Q|T_{D,syn_D})P(T_{Q,syn_Q}|T_{D,syn_D}) = \sum_{(syn_D, syn_Q) \in A} \frac{1}{N} \cdot P(T_{Q,syn_Q}|T_{D,syn_D}) \quad (6)$$

awhere N is the number of elements in the set A.

T_{D,syn_D} and T_{Q,syn_Q} represent a set of dependent terms having syn_D and syn_Q in a document and a query, respectively. For example, $T_{Q,parent-child}$ of the query in Figure 4 are (anthrax,sarin), (anthrax,sarin), (anthrax,sarin), (anthrax,ricin), (anthrax,soman), and (chemical,toxic). The probability $P(T_{Q,syn_Q}|A, T_{D,syn_D})$ is computed for each term pair that has a syntactic relation syn_Q in a query and syn_D in a document as follows:

$$P(T_{Q,syn_Q}|syn_D, syn_Q, T_{D,syn_D}) = \prod_{(t_i,t_j) \in T_{Q,syn_Q}} \lambda(t_i, t_j)P(t_i, t_j|T_{D,syn_D}) \quad (7)$$

where t_i and t_j are dependent terms with relations syn_Q in the parse tree of a query. Because the model considers more complex term dependencies, a harmful term dependency is more frequently introduced by unimportant terms in a query. To circumvent this problem, we use the query term ranking score of [18]. $\lambda(t_i, t_j)$ is the mean value of query term ranking scores of t_i and t_j. Each probability is calculated in the same way of the potential function in [16].

$$P(t_i, t_j|T_{D,syn_D}) = (1 - \alpha_D)\frac{tf_{t_i,t_j,syn_D}}{|D|} + \alpha_D \frac{cf_{t_i,t_j,syn_D}}{|C|} \quad (8)$$

tf_{t_i,t_j,syn_D} is the term frequency of term pairs t_i and t_j with the syntactic relation syn_D in a document D. We smoothed the probability distribution using the Dirichlet smoothing algorithm. cf_{t_i,t_j,syn_D} is the collection frequency of term pairs t_i and t_j with the syntactic relation syn_D in a collection.

Compared to the quasi-synchronous models for machine translation, QA and paraphrasing in the previous work [21, 6, 26], our quasi-synchronous model has three different characteristics: (1) Our model aims to align between a sentence and a document. (2) The model only consider syntactic variations through a quasi-synchronous stochastic process. (3) The model covers the four syntactic relations not only in a document but also in a query.

First, basically, our model matches terms of different units: a document and a query. On the other hand, in the previous work, matching is conducted between sentences. Therefore, terms of a query are allowed to match multiple times with terms in a document in information retrieval. As pointed out by Wang et al. [26], we are not interested in actual alignment between a query and a document. Rather, we interpret the process of matching as one in which a term pair with various dependency relations is generated by a document.

Second, our model focuses only on syntactic variations between queries and documents. The original quasi-synchronous grammar is established to consider both syntactic and semantic variations. Smith and Eisner[21] allows the original quasi-synchronous model to have a NULL alignment and a 1-to-2 alignments. Wang et al.[26] extends a term pair in a question to consider $2^{13} - 1$ possible combinations of expanded term pairs by using a thesaurus, WordNet. A quasi-synchronous model for information retrieval must take into account an entire collection while previous work has targeted a relatively small number of sentences. It is impractical to consider these kinds of semantic variations for all documents. Therefore, we make the model consider only term pairs which are lexically identical. We also ignore the transformations to a NULL node, an identical node and an arbitrary term pair in a sentence.

Third, most of the previous works [7, 13, 21, 22] treat only a parent-child relation or a head-modifier dependency in the parse tree of a query. On the other hand, in the

What is the prognosis *for* new drugs?
Find *ways of* measuring creativity.
What are commercial *uses of* Magnetic Levitation?
Mexico City *has the worst* air pollution *in the* world
What are the arguments *for and against* Great Britain's approval *of* women *being* ordained *as* Church *of* England priests?
What are the industrial *or* commercial uses *of* cyanide *or its* derivatives?

Figure 5: Example queries from the Robust 2004 collection which demonstrate better results when to assign more weight to the query likelihood model, the sequential dependence model and the quasi-synchronous model, respectively.

quasi-synchronous model, we expect to cover various dependency relations of terms in both a query and a document. As all the four syntactic configurations in a document can have important meanings, dependent terms having the four syntactic configurations in a query would also be important to find relevant documents. Thus, we model the transformation from all the four syntactic configurations to all the four syntactic configurations instead of counting only the direct head-modifier relation.

3.2 Parameter Optimization

Although the quasi-synchronous model is based on the premise that syntactically dependent terms are important for retrieving relevant documents, a retrieval model based on independence assumptions or simpler dependence assumptions is also effective and may outperform our model for at least some queries. The quasi-synchronous model ranks a document based on complex syntactic term dependencies in a parsing tree, so it is less applicable for a query containing a single important keyword. Figure 5 shows some example queries. In the first group of queries, the important terms are not expected to be used with specific dependencies. Therefore, regarding terms individually is sufficient to retrieve relevant documents. Because dependent terms in the second group of queries are placed near each other, the quasi-synchronous model may considers unnecessary dependencies and assigns inaccurate scores to the documents containing these dependencies. We need to combine the quasi-synchronous model with other retrieval models to address this problem.

When we combine several retrieval models, it is important to assign a proper weight to each retrieval model according to the characteristics of queries. Metzler [15] claimed that applying different types of models to new tasks typically requires an information retrieval expert to modify the underlying model in some way to properly account for the new types of features. He suggests an automatic feature selection model which determines the optimal weight of a linear feature-based model by using a greedy procedure. Zhai and Lafferty [29] also stressed an optimal parameter setting depends upon not only on document collections but also queries. Consider a linear interpolated the quasi-synchronous model with the sequential dependence model as follows:

$$P(q,d) = \lambda \cdot SDM(q,d) + (1-\lambda) \cdot QuasiSync(q,d) \quad (9)$$

where $SDM(q,d)$ and $QuasiSync(q,d)$ represent the log-scores of the document d given a query q measured by the sequential dependence model and quasi-synchronous model, respectively. In this preliminary experiment, we select an optimal parameter λ value which maximizes the overall average precision of the interpolated retrieval model. Figure 6 demonstrates that the distribution of weights according to the length of queries. This result demonstrates that the sequential dependence model and the quasi-synchronous model have their own advantages for different types of queries. If we use a fixed parameter for all queries, the quasi-synchronous model improves the effectiveness for some queries but also adversely affects the performance for other queries.

We can find an optimal parameter setting for a new task and document collection although this may require excessive tuning. On the other hand, it is impossible to optimize a parameter setting for an unseen query. To solve this problem, we exploit a machine learning approach to predict the optimal parameter settings for individual retrieval models based on a given query. Machine learning methods have been intensively studied for query term ranking approaches to predict the importance of an individual term or a set of terms in a given query [1, 2, 10, 11, 18, 28]. We extend this general approach to weighting the combination of different retrieval models.

In training data for training a query term ranking model, each term or the set of terms is labeled by its optimal weights. Similarly, we train a prediction model to measure the importance of a individual retrieval model for a given query. Training data for the prediction model consists of a query and the optimal weights of retrieval models,

$$(x_1, w_{i1}), \ldots, (x_n, w_{in})$$

where x_j is a ith query and its feature vector. w_{ij} is the

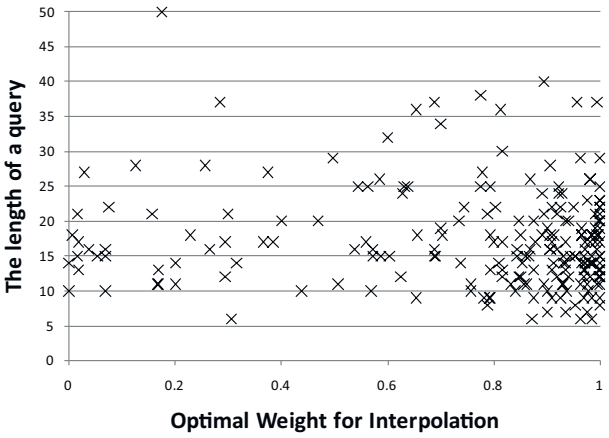


Figure 6: The distribution of optimal weight according to the length of a query from Robust 2004 collection when linear interpolating the sequential dependence model and the quasi-synchronous model.

optimal parameter setting of a jth sub retrieval model for the ith query.

A training label w_{ij}, which is the optimal weight for a ith query and a jth retrieval model, are selected empirically. First, we retrieve an initial ranked list of documents for a specific query by using a baseline retrieval model. When we retrieve initial documents, we make the retrieval model retrieve more documents because we wish training data cover documents out of the rank which may be retrieved by different parameter settings. Then, we choose optimal parameter values of retrieval models which maximize the performance of the initial document set. In this paper, we used mean average precision (MAP) as the retrieval metric. The weights in Figure 6 was chosen in this way.

The regression model we adopt in this paper is Support Vector Regression (SVR) [4]. Table 1 shows the list of features. Statistical features are an aggregation of feature values representing the characteristics of an individual term. Syntactic features are derived from the dependency parsing result of a query. These features are used to measure the number of the complex syntactic relations of terms in the query have. The higher the number of noun phrases and the depth of a parsing tree, the more complex the syntactic structure of a query. The quasi-synchronous model aims to capture these complex syntactic relations of terms from both queries and documents. These features also reflect whether we need to consider syntactic variations between a query and a document. For example, if a query is a wh-question such as a factored question, a relevant document contains terms of the query without rephrasing it.

Table 1: A summary of features used to measure importance of models for an individual query.

Statistical Features	
q_len	The length of query
aver_TF/DF/TFIDF	The average of term frequency, document frequency and TFxIDF of terms in a query
ratio_NOUN/ADJ/VERB	The ratio of nouns, adjectives and verbs in a query per the query length
KeyConcept	The ratio of terms which is selected by query term ranking methods
mean_score	The average of query term ranking scores
stopwords	The ratio of stopwords in a query
Syntactic Features	
is_question	Is a query a question?
is_wh	Is a query a wh-question?
num_NP	The number of a noun phrases in a query
ratio_NP	The number of a noun phrases in a query per the query length
num_clause	The number of a dependent clauses in a query
ratio_clause	The number of a dependent clauses in a query per the query length
aver_depth	the average depth of Key-Concept terms in a parsing tree of a query
height_tree	The height of a parsing tree of a query
ratio_PC/AD / SB/CC	The ratio of dependent term pairs which have parent-child, ancestor-descendent, siblings and c-commanding relations in a query, respectively.

4. EXPERIMENTS

In this section, we describe experiments to evaluate the quasi-synchronous model. Especially, we aim to compare the effectiveness of the quasi-synchronous approach to the independence and sequential dependency assumptions. For this purpose, we interpolated the quasi-synchronous model with the query likelihood model [19] and the sequential dependent model [16] to compare the quasi-synchronous model in different settings.

4.1 Experimental Settings

We made use of the TREC Robust 2004 and Gov2 collections for experiments. Some statistics for these two collections are shown in Table 2. All documents were indexed using the Indri search engine [17]. Terms were stemmed by using the Porter stemmer and were stopped using a standard list of stopwords. Only the description queries were used because our approach targets queries submitted in well-formed sentences[1]. We used Dirichlet smoothing for both the quasi-synchronous model and other retrieval models.

Table 2: The statistics of the TREC Robust 2004 and Gov2 collections

Coll	Description	# Doc.	TREC topics
Ro04	Newwire articles	528,155	301-450, 601-700
Gov2	.gov web collection	25,205,179	701-850

For the Robust 2004 collection, all documents and queries were parsed using the Stanford dependency parser [9]. The Stanford dependency parser internally includes the Stanford Part-of-Speech tagger [24]. Because the quasi-synchronous model needs an acyclic tree structure, we use the basic dependency representation form instead of the Stanford parser's collapsed representation. For the Gov2 collection, it is impractical to parse all documents. Therefore, we retrieve an initial document set using a baseline retrieval model. We then parsed documents in the initial set and evaluate the quasi-synchronous model only on the initial document set. Because the dependency parser accepts raw text format, we used ***lynx***[2] to convert the documents of the Gov2 collection in the TREC web format to raw text format before parsing documents.

To predict optimal weights for the interpolation of retrieval models, we used the support vector regression method which predicts the approximate target value based on a given feature vector [4]. We trained the regression model for each query using leave-one-out cross-validation in which one query was used for test data and the others were used for training data. Among the features in Table 1, some features are not appropriate a certain retrieval model. For exam-

[1]Compared to previous work, we use original description queries instead of refined queries from which command stop phrases are manually deleted.

[2]http://en.wikipedia.org/wiki/Lynx_(web_browser)

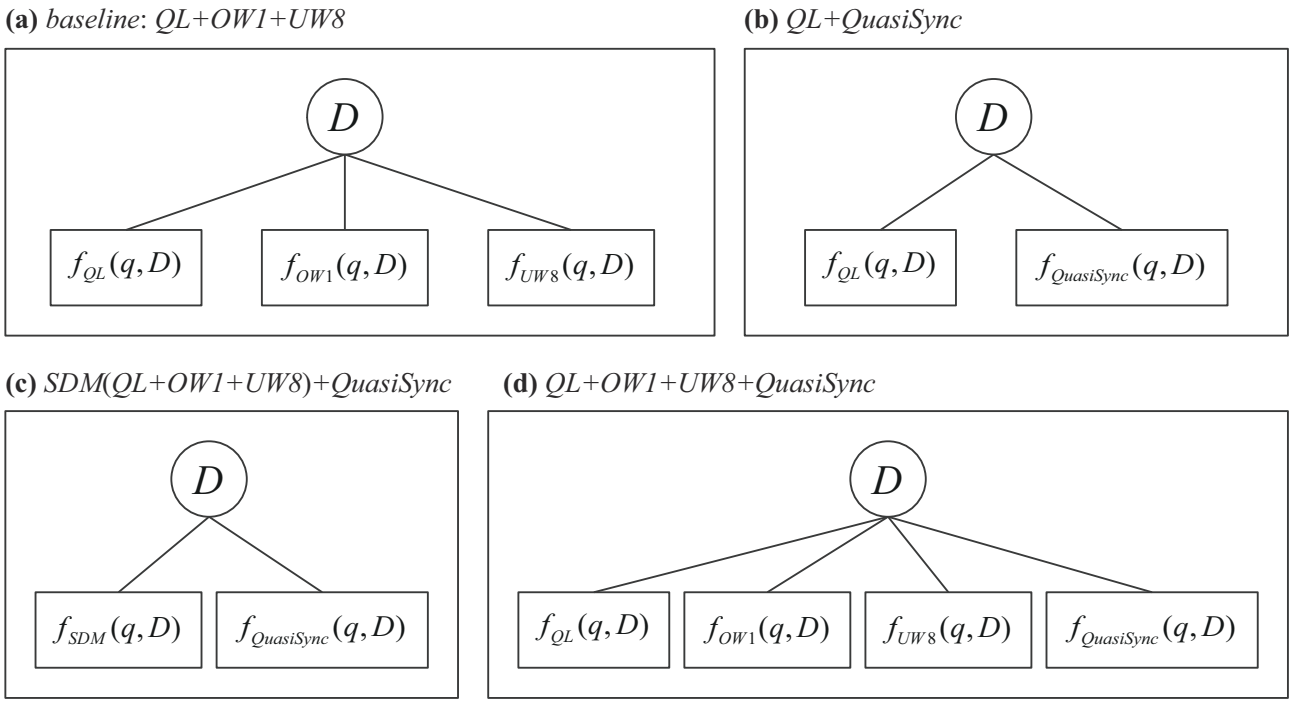


Figure 7: Four strategies of linear interpolation with the query-likelihood model(QL), the sequential dependence model(SDM), and the quasi-synchronous model(QM). The sequential dependence model interpolates three scores with fixed weights: the query-likelihood score f_{QL}, the ordered window score f_{OR1} and the unordered window score f_{UW8} [16].

ple, the number of term pairs having parent-child, ancestor-descendent, siblings and c-commanding in a query may be not discriminative to weight the ordered window function and the unordered window function in the sequential dependence model. Thus, we chose ten features for each interpolation strategy based on an self-evaluation result using the training data.

The dependent term pairs of the quasi-synchronous model include 98.31% of adjacent term pairs (3,365) Among the 3,365 adjacent term pairs, 54.65% (1,839) of them have a parent-child relation. 19.67% (662), 6.03% (203) and 19.64% (661) have ancestor-descendent, siblings and c-commanding relations, respectively.

4.2 Quasi-Synchronous Matching vs. Exact Matching

We combine the quasi-synchronous model with two baseline retrieval models using four different interpolation strategies. Figure 7 shows these four linear interpolation strategies. The first baseline model is the query-likelihood model (QL), a standard bag-of-word retrieval model based on the independent assumption [19]. The other baseline model is the sequential dependence model (SDM), which consists of three factors: a query likelihood factor, an ordered window factor and an unordered window factor [16]. In the four interpolation strategies, the three factors of the sequential dependence model are interpolated in two ways. The "$SDM + QuasiSync$" strategy interpolates three factors using fixed weights–$f_{SDM} = 0.85 \cdot f_{QL} + 0.10 \cdot f_{OW1} + 0.05 \cdot f_{UW8}$–and, then, interpolates f_{SDM} and $f_{QuasiSync}$ using predicted optimal weights. The "$QL+OW1+UW8+QuasiSync$" strategy use predicted weights for all the individual factors: f_{QL}, f_{OW1} and f_{UW8}.

Table 3 shows the experimental results of the four interpolation strategies. The statistical significance of the differences in the performance is determined using a two-sided Wilcoxon sign test, with $\alpha < 0.05$. As described in section 3, compared to the previous work, the quasi-synchronous model is different with respect to allowing inexact matching of syntactic relations between queries and documents. In Table 3, "**Quasi-Synchronous Matching**" represents the experimental results where we allow inexact matching while "**Exact Matching**" shows the experimental results when we align between term pairs having only a same syntactic relation. For all interpolation strategies, the quasi-synchronous approach shows better results than exact matching.

Among the four interpolation strategies in Figure 7, all the interpolation strategies with the quasi-synchronous model show significant improvements compared to a stat-of-art baseline model, the sequential dependence model. $SDM+QuasiSync$ achieves the best improvement. On the other hand, predicting the weights for the factors of the sequential dependence model fails to show improvement. The performance of $QL + OW1 + UW8$ is similar with that of the sequential dependence model. The performance of $QL + OW1 + UW8 + QuasiSync$ is slightly worse than that of $SDM + QuasiSync$.

4.3 Using True Optimal Parameters

To see the potential of the quasi-synchronous model, we

Table 3: Experimental results with the Robust 2004 with four interpolation strategies. Numbers in parentheses depict % improvement over the sequential dependence model.

	MAP	nDCG	prec@10
QL	0.2414	0.5061	0.4096
SDM	0.2477	0.5097	0.4217
$QL + OW1 + UW8$	0.2462 † (-0.62%)	0.5067 † (-0.59%)	0.4177 (-0.95%)
Quasi-Synchronous Matching			
$QL + QuasiSync$	0.2754 $^{\dagger}_{\star}$ (11.19%)	**0.5473**$^{\dagger}_{\star}$ (7.38%)	0.4606$^{\dagger}_{\star}$ (9.24%)
$SDM + QuasiSync$	**0.2786** $^{\dagger}_{\star}$ (12.48%)	0.5472$^{\dagger}_{\star}$ (7.37%)	**0.4614**$^{\dagger}_{\star}$ (9.43%)
$QL + OW1 + UW8$ $+QuasiSync$	0.2765$^{\dagger}_{\star}$ (11.61%)	0.5440$^{\dagger}_{\star}$ (6.74%)	0.4582$^{\dagger}_{\star}$ (8.67%)
Exact Matching			
$QL + Exact$	0.2553 $_{\star}$ (3.07%)	0.5231 $^{\dagger}_{\star}$ (2.63%)	0.4273 (1.33%)
$SDM + Exact$	0.2590 $^{\dagger}_{\star}$ (4.54%)	0.5234 $^{\dagger}_{\star}$ (2.70%)	0.4345 $_{\star}$ (3.05%)
$QL + OW1 + UW8$ $+Exact$	0.2583 $^{\dagger}_{\star}$ (4.27%)	0.5218 $^{\dagger}_{\star}$ (2.39%)	0.4361 $_{\star}$ (3.43%)

$\star$ denotes significantly different with QL
† denotes significantly different with SDM

Table 4: Mean Average Precision of the Robust 2004 collection when we use the four interpolation strategies using the true optimal weights of the training data. *Matching* **means "Quasi-Synchronous Matching" or "Exact Matching" in the second and third column, respectively.**

	MAP	
	Quasi	**Exact**
SDM	0.2477	
$QL + OW1 + UW8$	0.2725	
$QL + Matching$	0.3013	0.2699
$SDM + Matching$	0.3022	0.2724
$QL + OW1 + UW8 + Matching$	**0.3165**	0.2936

evaluate the four interpolation strategies using the training label as the interpolation weights. Table 4 shows the experimental results with the Robust collection.

When we use the training label or the true optimal weight, the (QL+$OW1$+$UW8$+$QuasiSync$) strategy demonstrates the best results. The QL+$OW1$+$UW8$ strategy is also better than the baseline. This demonstrates that the sequential dependence model still has a considerable margin for being improved by using a proper parameter setting instead of a fixed parameter setting.

Comparing the MAP value of QL+$QuasiSync$ or SDM+$QuasiSync$ with QL+$OW1$+$UW8$, the quasi-synchronous model has higher potential for taking into an account term dependencies than the sequential dependency model. Meanwhile, (QL+$OW1$+$UW8$+$QuasiSync$) shows considerable improvement compared to $SDM + QuasiSync$. $SDM +$ $QuasiSync$ assigns the same weights to adjacent term pairs while ($QL + OW1 + UW8 + QuasiSync$) gives different weights based on the query. It means that certain types of dependency could prove superior for a given query. Thus, we expect further improvement by using a different probability distribution for the alignment $P(syn_D, syn_Q|T_{D,syn_D})$ in Eq. 6.

On the other hand, the exact matching approach fails to show the potential to improve the effectiveness of a retrieval model even though (QL+$OW1$+$UW8$+$QuasiSync$) shows a significant improvement over $QL + OW1 + UW8$. The sequential dependence model can take account of long-distance term dependencies on the document side by the unordered window factor $UW8$ [16] and the exact matching approach considers long-distance term dependencies on the query side by extracting dependent terms having a parent-child, ancestor-descendent, siblings or c-commanding relation. Because the exact matching approach does not consider the possibility of the transformation of dependency relations between queries and documents, the gap of MAP values between QL+$ExactMatching$ and SDM+$ExactMatching$ is bigger than that of QL+$QuasiSync$ and SDM+$QuasiSync$. Only QL+$OW1$+$UW8$+$ExactMatching$ achieves similar improvement to the quasi-synchronous model.

Table 5: Experimental results with the Gov2 collection based on an initial document set retrieved by the sequential dependence model. Numbers in parentheses depict % improvement in each evaluation measure.

	Gov2		
	MAP	**nDCG**	**P10**
SDM	0.2654	0.5234	0.5195
QL+OW1+UW8	0.2674 (0.75%)	0.5246 (0.22%)	0.5228 (0.65%)
$SDM + QuasiSync$	0.2755 † (3.81%)	**0.5352** † (2.25%)	**0.5443** (4.78%)
$QL + OW1 + UW8 +$ $QuasiSync$	**0.2764** † (4.14%)	0.5342 † (2.06%)	0.5396 (3.88%)

† means statistically significance difference with SDM

4.4 Experimental Results with Web Collection

We also applied the quasi-synchronous model to a web collection, the Gov2 collection. For the Gov2 collection, it is impractical to parse all documents. We retrieve an initial document set (1,000 documents) using the sequential dependence model and then run experiments against this initial document set.

Table 5 shows the experimental results for the Gov2 collection. The performances of the interpolation strategies do not show as much improvement as the Robust 2004 collection. Still, $SDM + QuasiSync$ and $QL + OW1 + UW8 +$ $QuasiSync$ strategies improve the effectiveness significantly.

4.5 Analysis By Query Length

Compared to the sequential dependence model, the quasi-synchronous model aims to capture long distance dependencies in queries. To test the impact of the quasi-synchronous model on long distance dependencies, we analyze queries for

Table 6: Comparison of the MAP of the sequential dependence model, SDM, and $SDM+QuasiSync$. Statistics are collected from the experiments with the Robust 2004 collection. *# queries* is the number of queries belong to each group and *query length* is the average length of queries.

	# queries	query length
$SDM \geq QL + OW1 + UW8$	163	17.12
$SDM < QL + OW1 + UW8$	86	16.78
$SDM \geq SDM + QuasiSync$	98	15.14
$SDM < SDM + QuasiSync$	151	18.21

which this quasi-synchronous model shows better or worse results. Table 6 demonstrates comparison results between SDM, $QL + OW1 + UW8$ and $SDM + QuasiSync$. The upper two rows are the comparison of the sequential dependence model with fixed and predicted weights. This result demonstrates that the length of queries does not matter for the sequential dependence model itself. On the other hand, the lower two rows are the comparison between the sequential dependence model, SDM, and $SDM+QuasiSync$. It shows that queries improved by the quasi-synchronous model tend to be longer than the other queries.

Based on this observation, we analyze the experimental results of the Robust 2004 collection according to the average length of queries. Table 7 shows an experimental result in which we compared MAP of queries classified according to the length of queries. In this experiment, queries are split into three groups according their length. In the table, the interpolation strategies having the quasi-synchronous model demonstrate a clear tendency to larger improvements for longer queries while the strategy $QL + OW1 + UW8$ does not. The longer a query is, the more long-distance term dependencies the quasi-synchronous model extracts from the query that are not considered by the sequential dependency model. These experimental results show that the quasi-synchronous model can help to capture long-distance term dependencies in not only documents but also queries.

5. CONCLUSIONS

We propose a novel term dependence model, the quasi-synchronous model, inspired by a quasi-synchronous stochastic process in which an inexact matching of syntactic relations between source and target sentences is permitted. Similar to query term expansion techniques that address lexical variation between queries and document, we aim to support syntactic divergence of term dependencies from documents to queries using an inexact matching approach. We generalize a quasi-synchronous stochastic process to an information retrieval tasks in which matching occurs between a sentence and a document. The experimental results show that the quasi-synchronous model can significantly improve effectiveness compared to a strong state-of-the-art retrieval model.

However, each retrieval model has its own strengths and weaknesses and they differ query-by-query. A simpler retrieval model may be superior to a more sophisticated model depending on a given query. This is why most previous work using term dependencies has had problems showing consistent improvement. To address this issue, we use a machine learning approach to find an optimal parameter setting for a combination of retrieval models. By using a predicated optimal weight, we can optimize the overall performance of the interpolation of several retrieval models. Our method using the quasi-synchronous model combined with other models outperforms a strong state-of-the-art baseline.

Table 7: Experimental results with the Robust 2004 according to the length of queries. *Length* is the number of terms in a query and *# queries* is the number of queries belonging to each group. Numbers in parentheses depict % improvement in each evaluation measure.

	Robust 2004 (MAP)		
Length	**$\sim$ 10**	**11 $\sim$ 20**	**21 $\sim$**
# queries	**43**	**147**	**59**
SDM	0.3062	0.2398	0.2248
$QL + OW1 + UW8$	0.3056 (-0.19%)	0.2380 (-0.75%)	0.2232 (-0.71%)
$QL + QuasiSync$	0.3268 (6.72%)	0.2613 (8.98%)	0.2731 (21.51%)
$SDM + QuasiSync$	0.3255 (6.29%)	0.2668 (11.27%)	0.2738 (21.81%)
$QL + OW1 + UW8 + QuasiSync$	0.3251 (6.17%)	0.2649 (10.47%)	0.2698 (20.04%)

6. FUTURE WORK

In this paper, we use uniform distributions alignment between different syntactic relations in queries and documents. However, intuitively, dependent terms are expected to be used less frequently in a certain syntactic configurations and more frequently in others. For example, dependent terms such as technical terminology, proper names, etc. will be used in the same way by both users and authors. Moreover, as shown in the experimental results, a certain syntactic configuration could prove more important for evaluating the relevance of documents. Thus, instead of a uniform distribution, employing a weighted alignment model could improve the effectiveness of the quasi-synchronous model.

7. ACKNOWLEDGEMENT

This work was supported in part by the Center for Intelligent Information Retrieval and in part by ARRA NSF IIS-9014442 . Any opinions, findings and conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsor.

8. REFERENCES

[1] M. Bendersky and W. B. Croft. Discovering key concepts in verbose queries. In *Proc. ACM SIGIR*, pages 491–498, 2008.

[2] M. Bendersky, D. Metzler, and W. B. Croft. Learning concept importance using a weighted dependence model. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 31–40, 2010.

[3] A. Berger and J. Lafferty. Information retrieval as statistical translation. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 222–229, 1999.

[4] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[5] W. B. Croft, H. Turtle, and D. Lewis. The use of phrases and structured queries in information retrieval. In *Proceedings of the 14th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 32–45, 1991.

[6] D. Das and N. A. Smith. Paraphrase identification as probabilistic quasi-synchronous recognition. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, pages 468–476, 2009.

[7] J. Gao, J. Y. Nie, G. Wu, and G. Cao. Dependence language model for information retrieval. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 170–177, 2004.

[8] D. Gupta and N. Chatterjee. Study of divergence for example based English-Hindi machine translation. In *Proc. STRANS*, pages 132–140, 2002.

[9] D. Klein and C. D. Manning. Accurate Unlexicalized Parsing. *Proceedings of the 41st Meeting of the Association for Computational Linguistics*, pages 423–430, 2003.

[10] G. Kumaran and V. Carvalho. Reducing long queries using query quality predictors. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 564–571, 2009.

[11] C. Lee, R. Chen, S. Kao, and P. Cheng. A term dependency-based approach for query terms ranking. In *Proc. CIKM*, pages 1267–1276, 2009.

[12] C. Lee, G. G. Lee, and M.-G. Jang. Dependency structure applied to language modeling for information retrieval. *ETRI journal*, 28(3):337–346, 2006.

[13] L. Maisonnasse, E. Gaussier, and J. P. Chevallet. Revisiting the dependence language model for information retrieval. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 695–696. ACM, 2007.

[14] C. Manning and H. Schütze. *Foundations of statistical natural language processing*. MIT Press, 1999.

[15] D. Metzler. Automatic feature selection in the markov random field model for information retrieval. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 253–262, 2007.

[16] D. Metzler and B. W. Croft. 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*, pages 472–479, 2005.

[17] D. Metzler, T. Strohman, H. Turtle, and W. B. Croft. *Indri at TREC 2004: Terabyte track*. 2004.

[18] J. H. Park and W. B. Croft. Query term ranking based on dependency parsing of verbose queries. In *Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 829–830, 2010.

[19] J. M. Pont and W. B. Croft. A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 275–281. ACM, 1998.

[20] S. M. Shieber and Y. Schabes. Synchronous tree-adjoining grammars. In *Proceedings of the 13th conference on Computational linguistics-Volume 3*, pages 253–258, 1990.

[21] D. A. Smith and J. Eisner. Quasi-synchronous grammars: Alignment by soft projection of syntactic dependencies. In *Proceedings of the Workshop on Statistical Machine Translation*, pages 23–30, 2006.

[22] Y. Song, K. Han, S. Kim, S. Park, and H. Rim. A novel retrieval approach reflecting variability of syntactic phrase representation. *Journal of Intelligent Information Systems*, 31(3):265–286, 2008.

[23] M. Srikanth and R. Srihari. Biterm language models for document retrieval. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 425–426, 2002.

[24] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 173–180, 2003.

[25] C. Van Rijsbergen. A theoretical basis for the use of co-occurrence data in information retrieval. *Journal of Documentation*, 33(2):106–119, 1993.

[26] M. Wang, N. A. Smith, and T. Mitamura. What is the Jeopardy model? A quasi-synchronous grammar for QA. In *Proceedings of the the 2007 Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 22–32, 2007.

[27] J. Xu and W. B. Croft. Query expansion using local and global document analysis. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 4–11, 1996.

[28] X. Xue, S. Huston, and W. B. Croft. Improving verbose queries using subset distribution. In *Proc. CIKM*, 2010.

[29] C. Zhai and J. Lafferty. Two-stage language models for information retrieval. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 49–56, 2002.