Syntactic Query Models for Restatement Retrieval

Niranjan Balasubramanian and James Allan

Center for Intelligent Information Retrieval
University of Massachusetts Amherst, Amherst MA 01003
{niranjan,allan}@cs.umass.edu

Abstract. We consider the problem of retrieving sentence level restatements. Formally, we define restatements as sentences that contain all or some subset of information present in a query sentence. Identifying restatements is useful for several applications such as multi-document summarization, document provenance, text reuse and novelty detection. Spurious partial matches and term dependence become important issues for restatement retrieval in these settings. To address these issues, we focus on query models that capture relative term importance and sequential term dependence. In this paper, we build query models using syntactic information such as subject-verb-objects and phrases. Our experimental results on two different collections show that syntactic query models are consistently more effective than purely statistical alternatives.

1 Introduction

We describe and evaluate the task of finding restatements – sentences that match a query sentence, either in part or entirely. That is, starting from some sentence as a query, we define other sentences as relevant if they describe some or all of the same information units. Identifying sentences that contain overlapping information is a key challenge for several language applications such as tracking text reuse, summarization and novelty detection. Tracking information flow [14] and local text reuse [18] studied in the context of information provenance and plagiarism detection, focus on identifying sentence level information overlap. Also, extractive summarization techniques and novelty-based ranking often measure redundancy across sentences to avoid repeating information [1],[17],[23].

Cast as a sentence retrieval problem, the main challenge for restatement retrieval is that the query sentences are long: they usually contain multiple units of information (clauses), each of which could effectively be a query on its own. To effectively handle long queries, retrieval techniques must be able to identify key components of a query sentence [10],[11] and avoid spurious partial matches – accidentally flagging a match because a sentence includes partial information from different clauses. For document retrieval using keyword queries, the popular query likelihood (QL) model utilizes the frequency of query terms in the document, the length of the document and the collection frequency of the query terms. However, for sentence retrieval using long queries, the frequency of query

J. Karlgren, J. Tarhio, and H. Hyyrö (Eds.): SPIRE 2009, LNCS 5721, pp. 143–155, 2009.

terms in a sentence or in the collection does not provide adequate information to discriminate between relevant and spurious partial matches. Table 1 shows examples of spurious and relevant partial matches. Figure 1 illustrates the partial matches problem for query likelihood (QL) retrieval. At the top of the ranked list (the left of the figure) we have complete restatements – sentences which include all information units in the query and which tend to be identical to the query. However, as we go down the ranked list of sentences (to the right), we can see that there are several cases where the non-relevant sentences match more query terms than the relevant sentences.

Table 1. Partial Matches and Term Dependence: Square brackets enclose noun phrases with modifiers. Italicized words in the partial matches indicate query term matches.

Query Sentence *[Jordanian security officials]* on Sunday, announced the arrest of an *[Iraqi woman]*, closely linked to the *[terrorist leader Abu Musab al-Zarqawi]* as a *[fourth bomber]* in the *[Amman hotel attacks]* and they broadcast a *[taped confession]* showing her wearing a *[translucent suicide explosive belt]*, packed with *[ball bearings]* and describing how she had tried unsuccessfully to blow herself up.
Spurious partial match *Abu Musab al-Zarqawi* briefly survived the *bomb attack* that killed him Wednesday night, a military spokesman said Friday, *describing* the al-Qaida leader turning away and mumbling when American troops approached the stretcher that he had been placed on by *Iraqi* police officers.
Relevant partial match While a videotaped *confession showing* Rishawi *wearing* the disarmed *suicide belt* was being *broadcast* around the world, details about her life, motivation and role in the *attacks* that killed 57 people began to emerge in *Jordan* and *Iraq*.

Also, term independence assumptions can add to the partial matches problem. Consider the query shown in Table 1. Under term independence assumptions, non-relevant sentences that match many words in the pharases *Jordanian security officials* or *the terrorist leader, Abu Musab Al-Zarqawi* can receive higher query likelihood scores compared to relevant sentences which only match parts of these phrases. Addressing sequential term dependence – i.e., finding runs of adjacent terms that should be treated as a unit – can also help capture entities and concepts, moving us closer to semantic matching without actually trying to handle the semantics.

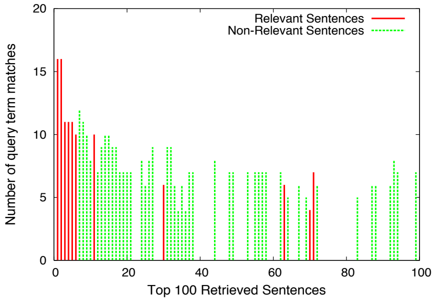


Fig. 1. Number of term matches for relevant and non-relevant sentences for an example query. Term matches for unjudged sentences in top 100 ranks are not shown.

In practice, spurious partial matches can lead to a large number of false alarms for identifying local text reuse. In extractive summarization and novelty detection, related but different events and news stories often use overlapping vocabulary inducing spurious partial matches that affect coverage of new information. In this paper, we investigate the partial term matches and term dependence problems in the restatement retrieval setting. In a sentence retrieval framework, we show that query models that emphasize the relative importance of query terms and capture sequential term dependence can improve retrieval effectiveness for our task. We compare syntactic query models to purely statistical alternatives and show that the syntactic models are more effective for restatement retrieval.

2 Related Work

The TREC Novelty Detection track focused on retrieval of novel on-topic sentences. The novelty detection techniques cannot be readily inverted to retrieve restatements; the typically poor precision of novelty detection [20], will lead to poor recall in retrieving restatements, especially the partial restatements. On the other hand, improvement in redundant information detection can help novelty detection by improving precision. Also, systems that used syntactic and semantic features for novelty detection improved precision. In a similar fashion, we believe that using syntactic features will help improve restatement retrieval.

Textual entailment is defined as the task of determining whether a given sentence entails an hypothesis [7]. Successful entailment systems often extract syntactic and semantic features from sentences and candidate hypotheses and utilize machine learning techniques to verify entailment [8]. While this task is similar to finding restatements, we focus on finding explicit restatements which are a subset of sentences that can entail a hypothesis. Moreover, the query sentences that we are interested in often contain multiple units of information unlike the sentences used for the entailment tasks.

In the context of document retrieval, Allan et al [2] studied the effect of extracting key components of TREC style queries and more recently, Bendersky et al [4] showed the utility of identifying key concepts in verbose description queries using a supervised learning approach to detect concepts – noun phrases – and weight them. Similarly, we believe the use of syntactic information will yield more effective query models for sentence retrieval.

For sentence matching, Metzler et al [14] showed that simple query likelihood (QL) model outperformed other word overlap based techniques such as TF-IDF for identifying sentences at various levels of similarity. Murdock [16] proposed translation based models for identifying sentence level similarity. Balasubramanian et al [3] showed that advanced language modeling techniques such as dependence models and relevance models can be combined to provide significant improvements over query likelihood baselines for finding redundant information. For sentence retrieval, Cai et al [5] used parse tree based features of candidate sentences to model term dependence in keyword queries for retrieving topically related sentences. In addition to using syntactic dependencies, we also use the subject-verb-object information to build query models and avoid parsing candidate sentences.

3 Query Models

Statistical approaches such as relevance models (RM) [12] and sequential dependence models (SDM) [15], can be used to model term importance and term dependence, respectively. For sentence retrieval, Balasubramanian et al [3] used relevance models based on term frequencies in source documents and showed that a combination model (DMRM) using dependence model queries to perform initial retrieval and then building relevance model queries provides additional improvements over either method.

However, there are some known issues with these purely statistical approaches. First, relevance models add new words to the query and rely upon a good quality initial retrieval. Instead of relying on an initial retrieval to build query models, syntactic analysis can be used to directly estimate the relative importance of terms. Furthermore, a better query model will improve the quality of the initial results and in-turn improve the quality of the relevance models. Second, term dependencies often span multiple terms and for long queries a full dependence model does not scale. Also, the simple sequential term dependence model is a brute force enumeration of all possible pairs of adjacent query terms. For keyword queries in document retrieval, this indiscriminate enumeration does not result in too many spurious matches. However, for long query sentences, it can cause spurious matches adding to the partial matches problem. Using syntactic dependencies we can avoid the problem of brute force enumeration of spurious dependencies.

The query sentences that we consider are well formed grammatical sentences that are amenable to automatic natural language analysis such as parsing techniques [6, 13, 22]. One approach to finding restatements would be to rank candidate sentences by measuring the alignment of their parse trees with that of the query sentence. However, this approach involves parsing entire text collections and performing computationally intensive alignment and is less robust since parsing can sometimes fail on candidate sentences. We propose to parse the query sentences alone to build effective query models for sentence retrieval. Thus, we leverage the benefits of syntactic information through parsing and the robustness of a retrieval framework to improve effectiveness of sentence matching in an efficient manner.

4 Syntactic Query Models

We use syntactic information to build query models for query likelihood retrieval. Specifically, we propose three syntactic query models to a] emphasize the relative importance of terms and b] capture the syntactic dependencies between query terms. We also build a combination model that leverages the benefits of these two models. Given a query sentence, we first parse the sentence using the Stanford Dependence parser [13] to obtain the syntactic dependencies. Then, we create *extended noun phrases* by grouping noun phrases, their adjectival modifiers and basic dependencies. We experimented with other types of modifiers and dependencies but found the adjectival modifiers to be most useful.

***SVO Weighting* (SW)**

Given a query sentence Q, we identify the *extended noun phrases* and words $\{c_1, c_2, \cdots, c_m\} \in Q$, and extract the set of subjects and objects, the verbs and the remaining phrases. Then, we formulate the query weights as follows:

$$\forall c \in Q, w(c) = \begin{cases} \delta(|c|) + \lambda_{ph} & \text{if c} \in \text{Phrases(Q)} \\ \delta(|c|) + \lambda_{so} & \text{if c} \in \text{Subj-Objs(Q)} \\ \lambda_v & \text{if c} \in \text{Verbs(Q)} \\ 1.0 & \text{otherwise} \end{cases} \tag{1}$$

where $|c|$ is the number of words in unit c. For our experiments we chose a δ to be a simple increasing function, $\delta(n) = 0.1n$. Finally, the weighted Indri query [21] is composed as follows:

$$SW(Q) = \#weight(w(c_1)\ c_1\ w(c_2)\ c_2\ \cdots\ w(c_m)\ c_m) \tag{2}$$

Initially, we found that even a fixed choice of values greater than 1 for the λs provides improvements. However, learning the parameters from a corpus allows us to tune the weights on the single word verbs in relation to the subjects, objects and other phrases which often span multiple words.

***Syntactic Phrase Matching* (SD)**

We use the syntactic dependencies obtained by parsing to effectively model sequential term dependencies of varying lengths. We create term dependence queries by modifying the sequential dependence model query described by Metzler et al [15]. Given the *extended noun phrases* extracted from the query sentence, $\{c_1, c_2, \cdots, c_m\}$ where $c_i = \{q_1^i, q_2^i, \cdots\}$, we construct an unordered window query for each extended noun phrase in the query and interpolate it with the original query sentence as follows:

$$SD(Q) = \beta(\mathrm{Q}) \cup (1-\beta) \left\{ \begin{array}{l} \#uw\delta_1(c_1) \\ \#uw\delta_2(c_2) \\ \cdots \\ \#uw\delta_m(c_m) \end{array} \right\} \tag{3}$$

where $\delta_i = |c_i| + 2$. $\#uw\delta(c)$ represents an un-ordered window query which estimates the likelihood of the words in the phrase c, occurring within a window of length δ. Figure 3 shows example Indri queries for the syntactic query models.

Syntactic Sub-Queries (SSQ)

We can extend the term dependence to larger units of information contained in the query sentence. We analyzed the parse trees of some query sentences and

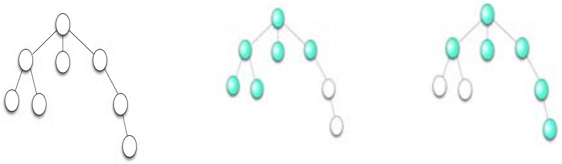
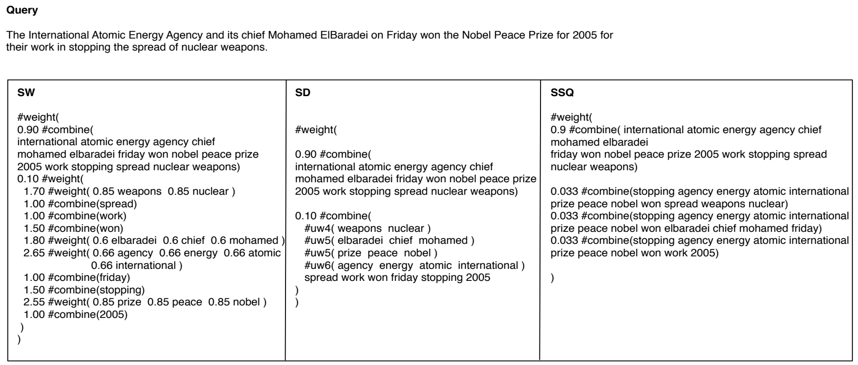


Fig. 2. Sub-queries: Tree on the left is an example parse of a query sentence. Colored nodes in the middle and right trees indicate words in the two possible sub-queries.



Query

The International Atomic Energy Agency and its chief Mohamed ElBaradei on Friday won the Nobel Peace Prize for 2005 for their work in stopping the spread of nuclear weapons.

SW

```
#weight(
0.90 #combine(
international atomic energy agency chief
mohamed elbaradei friday won nobel peace prize
2005 work stopping spread nuclear weapons)
0.10 #weight(
  1.70 #weight( 0.85 weapons  0.85 nuclear )
  1.00 #combine(spread)
  1.00 #combine(work)
  1.50 #combine(won)
  1.80 #weight( 0.6 elbaradei  0.6 chief  0.6 mohamed )
  2.65 #weight( 0.66 agency  0.66 energy  0.66 atomic
                 0.66 international )
  1.00 #combine(friday)
  1.50 #combine(stopping)
  2.55 #weight( 0.85 prize  0.85 peace  0.85 nobel )
  1.00 #combine(2005)
 )
)
```

SD

```
#weight(

0.90 #combine(
international atomic energy agency chief
mohamed elbaradei friday won nobel peace prize
2005 work stopping spread nuclear weapons)

0.10 #combine(
   #uw4( weapons  nuclear )
   #uw5( elbaradei  chief  mohamed )
   #uw5( prize  peace  nobel )
   #uw6( agency  energy  atomic  international )
   spread work won friday stopping 2005
)
)
```

SSQ

```
#weight(
0.9 #combine( international atomic energy agency chief
mohamed elbaradei
friday won nobel peace prize 2005 work stopping spread
nuclear weapons)

0.033 #combine(stopping agency energy atomic international
prize peace nobel won spread weapons nuclear)
0.033 #combine(stopping agency energy atomic international
prize peace nobel won elbaradei chief mohamed friday)
0.033 #combine(stopping agency energy atomic international
prize peace nobel won work 2005)

)
```

Fig. 3. Example Syntactic Query Models

observed that the root and the first level of the parse tree often contained information central to the query sentence. The different units seemed to correspond to the different sub-trees of the nodes in the first level. Therefore, for each query we generate a list of sub-queries one for each node in first level of the parse tree as follows: Extract nodes from the node's sub-tree and add them to the core of the query (the root and all the nodes in the first level). Figure 2 shows the sub-queries generated by our heuristic from a parse.

Combination Models (SWD and SWD-RM)

We experimented with various combinations of the query models. A linear interpolation of SW and SD models turned out to be the most useful combination and we refer to this model as *Syntactic Combination* (SWD). We also created a combination of the *Syntactic Combination* and the relevance model queries (SWD-RM) by using *Syntactic Combination* for initial retrieval and adding the *Syntactic Combination* query to the relevance model query, similar to the combination of dependence and relevance models (DMRM) [3]. This combination will demonstrate the utility of combining syntactic and purely statistical methods.

5 Experiments

To evaluate restatement retrieval we constructed a test bed from english newswire documents. First, we extracted 50 query sentences from the collection. The query sentences were chosen such that a) they could be potential responses to some actual query and b) many query sentences contained multiple units of information. Next, two annotators were asked to construct their own queries and find as many restatements as possible for each query sentence. Then, the restatements found in the first stage were used as surrogate queries and the results of a query likelihood sentence retrieval on the original query and the surrogate queries were used to create a pool of sentences for judging. Finally, we removed *easy* queries with QL MAP of 1.0 for the partial restatements category and *hard* queries that have no relevant judgments for the complete restatements category. Table 2 summarizes the details of the resulting collection (*Restatements Collection*). We also report results on a test collection consisting of 49 queries created for identifying redundant information [3](*Redundant Information Collection*).

Table 2. Collection Details

Collection	Documents	Sentences	Queries	Judged sentences	Restatements
English Newswire	79210	2,849,683	35	4340	647

We compare our syntactic approaches to statistical alternatives for building query models including the query likelihood baseline (QL) and the state-of-art document retrieval models – relevance models (RM) and sequential dependence models (DM). We also compare our combination model (SWD-RM) to the combination of relevance models and dependence models (DMRM) [3]. To train parameters for the query models, we performed grid search using cross-validation. For the Restatements collection we performed 5-fold cross validation and for the Redundant information collection we used 7-fold cross validation. Table 3 shows the training parameters for the various methods. DM and *Syntactic Phrase Matching* were tuned to optimize precision@10 to gain better generalization performance over MAP.

Table 3. Training parameters: a) λ - JM smoothing parameter b) β - Original query interpolation weight c) λ_{ph} - Phrase weight d) λ_{so} - Subject/Object weight e) λ_v - Verb weight f) γ_{svo} - SVO Query weight g) γ_{ph} - Syntactic Phrase Query weight h) fbt, fbd: Feedback terms and documents. i) w - interpolation weight.

QL	DM	RM	DMRM	SW	SD	SWD	SWD-RM
λ	λ, β	λ, β fbt, fbd	DM, RM + w	$\lambda, \beta, \lambda_{sv}, \lambda_v, \lambda_{ph}$	λ, β	$\lambda, \beta, \gamma_{svo}, \gamma_{ph}$	SW,SD+w

6 Results

6.1 Restatements collection

Table 4 shows results for the statistical and syntactic query models. The syntactic query models are consistently better than the statistical alternatives except for the *Syntactic Sub Queries* .

Table 4. 5-fold Cross Validation results on Restatements collection - Sig entries indicate significant improvements over the corresponding methods using paired t-test. (q:QL, r:RM, d:DM, dr:DMRM, sw:SW,sd:SD). Underlines indicate the best score.

Method	**P@5**	**P@10**	**P@20**	**MAP**	**Sig. (in MAP)**
QL	0.7086	0.5686	0.4514	0.6373	
DM	0.7086	0.5743	0.4429	0.6407	
RM	0.7086	0.5657	0.4443	0.6374	
DMRM	0.7200	0.5857	0.4500	0.6509	q,r
SW	0.7371	0.5971	0.4614	0.6660	q
SD	0.7200	0.5771	0.4471	0.6564	q
SWD	$\underline{0.7543}$	0.6000	$\underline{0.4629}$	0.6732	q,d,r
SSQ	0.7086	0.5714	0.4500	0.6358	
SWD-RM	0.7486	$\underline{0.6057}$	0.4586	$\underline{0.6788}$	q,d,r,dr, SW, SD

Statistical Query Models. In contrast to their known value for document retrieval, RM and DM do not provide any significant improvements in MAP over the query likelihood baseline. We observed that RM roughly helped half the queries and was detrimental to the other half. In addition to failures due to poor retrieval, even when baseline retrieval effectiveness was high, relevance models can add words from sentences that are only partial restatements of the original query sentence. This actually worsens the partial matches problem and lowers the overall effectiveness. For the DM, the poorly performing queries often overemphasize the match of a long noun phrase. For example, in the query fragment, *Iranian President Mahmoud Ahmadinejad*, the corresponding long noun phrase match *iran's president, mahmoud ahmadinejad* is amplified in the dependence model query by having multiple ordered window matches and a larger number of unordered window matches. The combination of dependence and relevance models, DMRM, provides good improvements over either model by leveraging their individual strengths [3]. Overall, the purely statistical approaches do not yield substantial improvements for restatement retrieval.

***SVO Weighting* and *Syntactic Phrase Matching*.** Intuitively, *SVO Weighting* weights the key components of the query sentence and balances the relative weights of these components. For query sentences with long noun phrases, the presence of the noun phrases alone can significantly boost the likelihood score of candidate sentences. Tuning the weights on phrases and verbs together provides a way to moderate the impact of matching long noun phrases alone and serves to improve the importance of verbs in candidate sentences. Table 5 displays a non-relevant sentence that matches the long noun phrase in the query sentence and some relevant sentences that only contain a portion of the long noun phrase but contain all the other key components of the query sentence. The difference in ranking according to dependence models and the *SVO Weighting* demonstrates the strength of the syntactic model in balancing the relative query term weights.

Syntactic Phrase Matching emphasizes the dependence between terms in the query sentence. By grouping noun phrases and their modifiers in an unordered

Table 5. Example illustrating benefits of SW compared to DM. Italicized words indicate term matches. DM and SW entries indicate ranks in corresponding methods.

Query: Iranian President Mahmoud Ahmadinejad called the holocaust a "myth"

DM	SW	Rel.	Text
21	38	NR	tel aviv, israel – with *iran's president, mahmoud ahmadinejad, call*ing for israel to be "wiped off the map," israeli officials have special reasons for concern now that iran has defied the west and said it will resume enriching uranium.
22	14	R	charlotte knobloch, president of the central council of jews in germany, noted that *ahmadinejad* has *call*ed the *holocaust* a "*myth*".
88	18	R	*ahmadinejad* produced outrage in the west last year when he threatened israel and *call*ed the holocaust a "*myth*".
97	19	R	*ahmadinejad* has generated international scorn by dismissing the *holocaust* as a *myth* and *call*ing for the destruction of israel.

window query it favors phrase matches thereby capturing entities and other important concepts in the query. Also, the syntactic grouping ensures that only valid dependencies are captured and spurious dependencies that are possible in a brute force enumeration are avoided. Compared to brute force DM queries, *Syntactic Phrase Matching* queries are more effective at modeling sequential term dependence and are also more efficient.

Syntactic Sub Queries. *Syntactic Sub Queries* did not provide any improvements over QL. This is mainly due to sub-queries that add only one or two unimportant words to the top level of the tree thus creating bad sub-queries that cause spurious matches. Furthermore, for some queries the top portion of the parse tree is not central to the query as assumed by our heuristic. However, we believe that by systematically capturing larger syntactic structures such as clauses, we can create more meaningful sub-queries.

***Syntactic Combination* Models.** The combination model SWD provides small improvements over either model used in the combination. However, the combination now consistently outperforms the DM and RM baselines. Similar to the combination method DMRM, the *SVO Weighting* and *Syntactic Phrase Matching* provide different types of evidences for relevance and their combination provides significant improvements. The SWD-RM provides minor improvements over SWD but it outperforms DMRM, *SVO Weighting* and *Syntactic Phrase Matching* thus showing that syntactic and statistical query models can be combined to obtain additional improvements. Finally, we note that the small improvements due to the combinations add up to an absolute 4 point improvement in MAP over the QL baseline.

6.2 Redundant Information

Table 6 shows results for this collection. The statistical query models follow the trends observed in [3]. Both *SVO Weighting* and *Syntactic Phrase Matching* outperform the QL baseline and DM significantly and the combination models SWD and SWD-RM significantly outperform all the statistical models except for DMRM. The results clearly confirm the trends we observed in our collection but the actual improvements are smaller. We believe that this is due to the differences in the types of queries in the collections. In comparison to Redundant information collection , the query sentences in Restatements collection are more complex, in terms of average length (17 versus 21 average words per query) and number of SVO elements. Overall, we expect the syntactic query models to benefit complex query sentences more. Although we did not observe any solid trends in terms of complexity versus the benefit of the syntactic methods, we observed that the benefits of using syntactic query models were limited in another noisy collection with simple queries.

To summarize, we see that utilizing syntactic information to build query models consistently outperforms purely statistical methods for term weighting and the brute force sequential dependence models.

Table 6. Redundant information - 5-fold Cross Validation results on 49 queries. Sig column entries indicate significant improvements in MAP over the corresponding methods (q:QL, r:RM, d:DM, dr:DMRM, sw:SW,sd:SD) using paired t-test.

Method	**P@5**	**P@10**	**P@20**	**MAP**	**Sig.**
QL	0.6122	0.5041	0.3500	0.5207	
DM	0.6204	0.4878	0.3510	0.5253	
RM	0.6245	0.5082	0.3500	0.5308	q,d
DMRM	0.6122	0.5041	0.3561	0.5279	
SW	0.6204	0.5143	0.3643	0.5338	q,d
SD	0.6327	0.5082	0.3622	0.5376	q,d
SWD	0.6380	0.5082	0.3622	0.5380	q,d,SW
SWD-RM	0.6286	0.5020	0.3643	0.5387	q,d,r,SW

7 Syntactic Features for Discriminative Training

We also conducted experiments on the Restatements collection to investigate the utility of syntactic features in a discriminative setting for sentence retrieval. Nallapati et al devised some language modeling based features for a discriminative classifier to rank documents. We adopted the same set of text based features (T) and investigated the use of parser based features such as, presence of subject, verb, object and distance based measures on the parse tree (P). Our initial results (see Table 7) shows that ranking and binary SVM [9] using text only features (T) perform comparably to that of the traditional retrieval approaches. More importantly, we see that adding the parser based features improves the performance of both ranking and binary SVMs similar to the improvements we observed with the syntactic query models. These results clearly show that there is potential for investigating syntactic features in a discriminative setting. We

Table 7. SVM based learning for restatement retrieval on Restatements collection

Method	P@5	P@10	P@20	MAP
RankSVM (T)	0.6971	0.5829	0.4514	0.6349
BinarySVM (T)	0.6914	0.5686	0.4443	0.6304
BinarySVM (P)	0.5143	0.3771	0.2774	0.3966
RankSVM (P + T)	0.7086	0.5857	0.4571	0.6520
BinarySVM (P+ T)	0.7314	0.5829	0.4586	0.6528

plan to extend our work to directly incorporate the syntactic dependencies and model more complex relationships amongst the features for ranking.

8 Conclusions

In this paper, we described the problem of partial term matches and term dependence for a general restatement retrieval task. We showed that query models that address the partial matches and sequential term dependencies, provide consistent gains in effectiveness. We find that syntactic query models are consistently more effective than purely statistical alternatives. Avoiding spurious partial matches is a key challenge for several natural language applications and syntactic query models provide an effective and efficient query dependent solution. Using syntactic query models, we leverage the benefits of both syntactic information as well as the robustness and efficiency of a retrieval framework by avoiding pitfalls due to parser failures and inaccuracies. Natural language applications often use syntactic and semantic features in a machine learning framework. Our initial experiments with a discriminative approach for retrieving restatements also shows promise for integration in such settings.

For long queries, indiscriminately expanding all query words can hurt effectiveness. We believe that syntactic information can lead to some selective expansion techniques that leverage the term importance and dependence in the query. Sentence simplification [19] techniques use syntactic information to break query and candidate sentences into simpler sentences. Another extension to our work is to consider sentence simplification for matching query and candidate sentences. Also, we intend to explore the effect of modeling the term dependencies more directly in a machine learning framework for restatement retrieval.

Acknowledgements

This work was supported in part by the Center for Intelligent Information Retrieval, in part by the Defense Advanced Research Projects Agency (DARPA) under contract number HR0011-06-C-0023, and in part by UpToDate. Any opinions, findings and conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsor.

References

1. Allan, J., Wade, C., Bolivar, A.: Retrieval and novelty detection at the sentence level. In: Proceedings of the 26th annual international ACM SIGIR conference, pp. 314–321 (2003)
2. Allan, J., Callan, J., Croft, B., Ballesteros, L., Broglio, J., Xu, J., Shu, H.: Inquery at TREC-5. In: Fifth Text REtrieval Conference (TREC-5), pp. 119–132 (1997)
3. Balasubramanian, N., Allan, J., Croft, W.B.: A comparison of sentence retrieval techniques. In: SIGIR 2007: Proceedings of the 30th annual international ACM SIGIR conference, pp. 813–814. ACM Press, New York (2007)
4. Bendersky, M., Croft, W.B.: Discovering Key Concepts in Verbose Queries. In: Proceedings of ACM SIGIR 2008 conference, pp. 491–498 (2008)
5. Cai, K., Chen, C., Liu, K., Bu, J., Huang, P.: MRF based approach for sentence retrieval. In: SIGIR 2007: Proceedings of the 30th annual international ACM SIGIR conference, pp. 795–796. ACM Press, New York (2007)
6. Charniak, E.: A maximum-entropy-inspired parser. In: ACM International Conference Proceeding Series, vol. 4, pp. 132–139 (2000)
7. Dagan, I., Glickman, O., Magnini, B.: The PASCAL Recognising Textual Entailment Challenge. In: Quiñonero-Candela, J., Dagan, I., Magnini, B., d'Alché-Buc, F. (eds.) MLCW 2005. LNCS (LNAI), vol. 3944, pp. 177–190. Springer, Heidelberg (2006)
8. Giampiccolo, D., Magnini, B., Dagan, I., Dolan, B.: The Third PASCAL Recognizing Textual Entailment Challenge. In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, pp. 1–9 (2007)
9. Joachims, T.: Making large scale SVM learning practical. Universität Dortmund (1999)
10. Kumaran, G., Allan, J.: A Case For Shorter Queries, and Helping Users Create Them. In: HLT 2007: NAACL Proceedings of the Main Conference, pp. 220–227 (2007)
11. Kumaran, G., Allan, J.: Effective and Efficient User Interaction for Long Queries. In: SIGIR 2008: Proceedings of the 31st annual international ACM SIGIR conference, pp. 11–18 (2008)
12. Lavrenko, V. and Croft, W.B.: Relevance based language models. In: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 120–127 (2001).
13. de Marneffe, M.C., MacCartney, B., Manning, C.D.: Generating typed dependency parses from phrase structure parses
14. Metzler, D., Bernstein, Y., Bruce Croft, W., Moffat, A., Zobel, J.: Similarity measures for tracking information flow. In: CIKM 2005: Proceedings of the 14th ACM international conference on Information and knowledge management, pp. 517–524. ACM, New York (2005)
15. Metzler, D., Croft, W.B.: A Markov random field model for term dependencies. In: SIGIR 2005: Proceedings of the 28th annual international ACM SIGIR conference, pp. 472–479 (2005)
16. Murdock, V.: Aspects of Sentence Retrieval. PhD Thesis. University of Massachussets Amherst (2006)
17. Radev, D., Jing, H., Stys, M., Tam, D.: Centroid-based summarization of multiple documents. In: Information Processing and Management, pp. 919–938 (2004)
18. Seo, J., Croft, W.B.: Local text reuse detection. In: Proceedings of the 31st annual international ACM SIGIR conference, pp. 571–578 (2008)

19. Siddharthan, A., Nenkova, A., McKeown, K.: Syntactic simplification for improving content selection in multi-document summarization. In: Proceedings of the COLING 2004, pp. 896–902 (2004)
20. Soboroff, I., Harman, D.: Novelty Detection: The TREC Experience. In: HLT/EMNLP, pp. 105–112 (2005)
21. Strohman, T., Metzler, D., Turtle, H., Croft, W.B.: Indri: A language model-based search engine for complex queries. In: Proceedings of the International conference on Intelligence Analysis (2004)
22. Taskar, B., Klein, D., Collins, M., Koller, D., Manning, C.: Max-margin parsing. In: Proc. EMNLP 2004, pp. 1–8 (2004)
23. Wang, D., Li, T., Zhu, S., Ding, C.: Multi-document summarization via sentence-level semantic analysis and symmetric matrix factorization. In: Proc. ACM SIGIR Conference, pp. 307–314 (2008)