Feature-Based Selection of Dependency Paths in Ad Hoc Information Retrieval

K. Tamsin Maxwell
School of Informatics
University of Edinburgh
Edinburgh EH8 9AB, UK
t.maxwell@ed.ac.uk

Jon Oberlander
School of Informatics
University of Edinburgh
Edinburgh EH8 9AB, UK
j.oberlander@ed.ac.uk

W. Bruce Croft
Dept. of Computer Science
University of Massachusetts
Amherst, MA 01003, USA
croft@cs.umass.edu

Abstract

Techniques that compare short text segments using dependency paths (or simply, paths) appear in a wide range of automated language processing applications including question answering (QA). However, few models in ad hoc information retrieval (IR) use paths for document ranking due to the prohibitive cost of parsing a retrieval collection. In this paper, we introduce a flexible notion of paths that describe chains of words on a dependency path. These chains, or catenae, are readily applied in standard IR models. Informative catenae are selected using supervised machine learning with linguistically informed features and compared to both non-linguistic terms and catenae selected heuristically with filters derived from work on paths. Automatically selected catenae of 1-2 words deliver significant performance gains on three TREC collections.

1 Introduction

In the past decade, an increasing number of techniques have used complex and effective syntactic and semantic features to determine the similarity, entailment or alignment between short texts. These approaches are motivated by the idea that sentence meaning can be flexibly captured by the syntactic and semantic relations between words, and encoded in dependency parse tree fragments. Dependency paths (or simply, paths) are compared using techniques such as tree edit distance (Punyakanok et al., 2004; Heilman and Smith, 2010), relation probability (Gao et al., 2004) and parse tree alignment (Wang et al., 2007; Park et al., 2011).

Much work on sentence similarity using dependency paths focuses on question answering (QA) where textual inference requires attention to linguistic detail. Dependency-based techniques can also be highly effective for ad hoc information retrieval (IR) (Park et al., 2011). However, few path-based methods have been explored for ad hoc IR, largely because parsing large document collections is computationally prohibitive.

In this paper, we explore a flexible application of dependency paths that overcomes this difficulty. We reduce paths to chains of words called *catenae* (Osborne and Groß, 2012) that capture salient semantic content in an underspecified manner. Catenae can be used as lexical units in a reformulated query to explicitly indicate important word relationships while retaining efficient and flexible proximity matching. Crucially, this does not require parsing documents. Moreover, catenae are compatible with a variety of existing IR models.

We hypothesize that catenae identify most units of salient knowledge in text. This is because they are a condition for ellipsis, in which salient knowledge can be successfully omitted from text (Osborne and Groß, 2012). To our knowledge, this paper is the first time that catenae are proposed as a means for term selection in IR, and where ellipsis is considered as a means for identification of semantic units.

We also extend previous work with development of a linguistically informed, supervised machine learning technique for selection of informative catenae. Previous heuristic filters for dependency paths (Lin and Pantel, 2001; Shen et al., 2005; Cui et al., 2005) can exclude informative relations. Alternatively, treating all paths as equally informative (Punyakanok et al., 2004; Park et al., 2011; Moschitti, 2008) can generate noisy word relations and is computationally intensive.

The challenge of path selection is that no explicit information in text indicates which paths are relevant. Consider the catenae captured by heuristic filters for the TREC[1] query, '*What role does blood-alcohol level play in automobile accident fatalities*' (#358, Table 1). It may appear obvious that the component words of '*role play*'

[1] Text REtrieval Conference, see http://trec.nist.gov/

Query: *What role does blood-alcohol level play in automobile* accident fatalities*? (*abbreviated to `auto', `fatal')*

Catenae	Governor-dependent	Predicate-argument	Nominal end slots	Sequential dependence
blood alcohol	blood alcohol			blood alcohol
level play	level play			level play
auto accident	auto accident	auto accident	auto accident	auto accident
accident fatal	accident fatal	accident fatal	accident fatal	accident fatal
role play	role play			
play fatal	play fatal	play fatal		
blood alcohol play				
play accident fatal		play accident fatal		
auto accident fatal		auto accident fatal	auto accident fatal	
level play fatal			level play fatal	
role play fatal			role play fatal	
role level play				
				role blood
				alcohol level
				play auto

Table 1: Catenae derived from dependency paths, as selected by heuristic methods. Selections are compared to sequential bigrams that use no linguistic knowledge.

and '*level play*' do not have an important semantic relationship relative to the query, yet these catenae are described by parent-child relations that are commonly used to filter paths in text processing applications. Alternative filters that avoid such trivial word combinations also omit descriptions of key entities such as '*blood alcohol*', and identify longer catenae that may be overly restrictive. These shortcomings suggest that an optimized selection process may improve performance of techniques that use dependency paths in ad hoc IR.

We identify three previously proposed selection methods, and compare them on the task of catenae selection for ad hoc IR. Selections are tested using three TREC collections: Robust04, WT10G, and GOV2. This provides a diverse platform for experiments. We also develop a linguistically informed machine learning technique for catenae selection that captures both key aspects of heuristic filters, and novel characteristics of catenae and paths. The basic idea is that selection, or weighting, of catenae can be improved by features that are *specific to paths*, rather than *generic for all terms*.

Results show that our selection method is more effective in identifying key catenae compared to previously proposed filters. Integration of the identified catenae in queries also improves IR effectiveness compared to a highly effective baseline that uses sequential bigrams with no linguistic knowledge. This model represents the obvious alternative to catenae for term selection in IR.

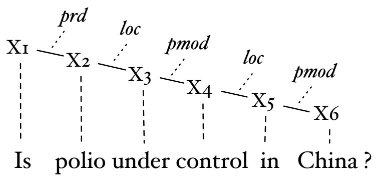
The rest of this paper is organised as follows. §2 reviews related work, §3 describes catenae and their linguistic motivation and §4 describes our selection method. §5 evaluates classification experiments using the supervised filter. §6 presents the results of experiments in ad hoc IR. Finally, §7 concludes the paper.

2 Related work

Techniques that compare short text segments using dependency paths are applied to a wide range of automated language processing tasks, including paraphrasing, summarization, entailment detection, QA, machine translation and the evaluation of word, phrase and sentence similarity. A generic approach uses a matching function to compare a dependency path between any two stemmed terms x and y in a sentence A with any dependency path between x and y in sentence B. The match score for A and B is computed over all dependency paths in A.

In QA this approach improves question representation, answer selection and answer ranking compared to methods that use bag-of-words and ngram features (Surdeanu et al., 2011). For example, Lin and Pantel (2001) present a method to derive paraphrasing rules for QA using analysis of paths that connect two nouns; Echihabi and Marcu (2003) align all paths in questions with trees for heuristically pruned answers; Cui et al. (2005) score answers using a variation of the IBM translation model 1; Wang et al. (2007) use quasi-synchronous translation to map all parent-child paths in a question to any path in an answer; and Moschitti (2008) explores syntactic and semantic kernels for QA classification.

In ad hoc IR, most models of term dependence use word co-occurrence and proximity (Song and Croft, 1999; Metzler and Croft, 2005; Srikanth and Srihari, 2002; van Rijsbergen, 1993). Syntactic language models for IR are a significant departure from this trend (Gao et al., 2004; Lee et al., 2006; Cai et al., 2007; Maisonnasse et al., 2007) that use dependency paths to address long-distance dependencies and normalize spurious differences in surface text. Paths are constrained in both



Catenae (stoplisted)	Dependency paths
polio	
polio control	polio $\xrightarrow{loc}$ under $\xrightarrow{pmod}$ control
control	
control China	control $\xrightarrow{loc}$ in $\xrightarrow{pmod}$ China
China	
polio control China	polio $\xrightarrow{loc}$ under $\xrightarrow{pmod}$ control $\xrightarrow{loc}$ in $\xrightarrow{pmod}$ China

Figure 1: Catenae are an economical and intuitive representation of dependency paths.

queries and documents to parent-child relations. In contrast, (Park et al., 2011) present a quasi-synchronous translation model for IR that does not limit paths. This is based on the observation that semantically related words have a variety of direct and indirect relations. All of these models require parsing of an entire document collection.

Techniques using dependency paths in both QA and ad hoc IR show promising results, but there is no clear understanding of which path constraints result in the greatest IR effectiveness. We directly compare selections of catenae as a simplified representation of paths.

In addition, a vast number of methods have been presented for term weighting and selection in ad hoc IR. Our supervised selection extends the successful method presented by Bendersky and Croft (2008) for selection and weighting of query noun phrases (NPs). It also extends work for determining the variability of governor-dependent pairs (Song et al., 2008). In contrast to this work, we apply linguistic features that are specific to catenae and dependency paths, and select among units containing more than two content-bearing words.

3 Catenae as semantic units

Catenae (Latin for '*chain*', singular *catena*) are dependency-based syntactic units. This section outlines their unique semantic properties.

A catena is defined on a dependency graph that has lexical nodes (or words) linked by binary asymmetrical relations called dependencies. Dependencies hold between a *governor* and a *dependent* and may be syntactic or semantic in nature (Nivre, 2005). A dependency graph is usually acyclic such that each node has only one governor, and one root node of the tree does not depend on any other node.

A catena is a word, or sequence of words that are continuous with respect to a walk on a dependency graph. For example, Fig. 1 shows a dependency parse that generates 21 catenae in total: (using i for Xi) 1, 2, 3, 4, 5, 6, 12, 23, 34, 45, 56, 123, 234, 345, 456, 1234, 2345, 3456, 12345, 23456, 123456. We process catenae to remove stop words on the INQUERY stoplist (Allan et al., 2000) and lexical units containing 18 TREC description stop words such as '*describe*'. This results in a reduced set of catenae as shown in Fig. 1.

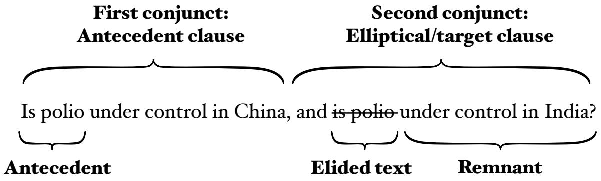


Figure 2: Ellipsis in a coordinated construct.

A dependency path is ordered and includes both word tokens and the relations between them. In contrast, a catena is a set of word types that may be ordered or partially ordered. A catena is an economical, intuitive lexical unit that corresponds to a dependency path and is argued to play an important role in syntax (Osborne et al., 2012).

In this paper, we explore catenae instead of paths for ad hoc IR due to their suitability for efficient IR models and flexible representation of language semantics. Specifically, we note that catenae identify words that can be omitted in elliptical constructions (Osborne et al., 2012). They thus represent *salient semantic information* in text. To clarify this insight, we briefly review catenae in ellipsis.

3.1 Semantic units in ellipsis

Fig. 2 shows terminology for the phenomenon of ellipsis. The omitted words are called *elided* text, and words that could be omitted, but are not, we call *elliptical candidates*.

Ellipsis relies on the logical structure of a coordinated construction in which two or more elements, such as sentences, are joined by a conjunctive word or phrase such as '*and*' or '*more than*'. A coordinated structure is required because the omitted words are 'filled in' by assuming a parallel relation p between the first and second conjunct. In ellipsis, p is omitted and its arguments are retained in text. In order for ellipsis to be successful and grammatically correct, p must be salient shared knowledge at the time of communication (Prince, 1986; Steedman, 1990). If p is salient then the omitted text can be inferred. If p is not salient then the omission of words merely results in ungrammatical, or incoherent, sentences.

This framework is practically illustrated in Fig.

	Ellided sentences	**Ellipsis candidates marked in italics: they are catenae**
	Is polio under control in China, and...	
a)	in India ?	Is polio under control in China, and (*is polio under control*) in India ?
b)	is cancer under observation ?	Is polio under control in China, and is cancer under observation (*in China*) ?
c) *	cancer observation ?	* Is polio under control in China, and (*is*) cancer (*under*) observation (*in China*) ?
d) *	under India ?	* Is polio under control in China, and (*is polio*) under (*control in*) India ?

Figure 3: For ellipsis to be successful, elided words must be catenae. Ellipsis candidates are catenae[2].

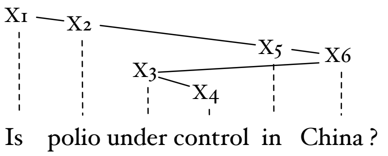


Figure 4: A parse in which '*polio China*' is a catena.

3 for the query, '*Is polio under control in China?*'. Sentences marked by * are incoherent, and it is evident that the omitted words do not form a salient semantic unit. They also do not form catenae. In contrast, the omitted words in successful ellipsis do form catenae, and they represent informative word combinations with respect to the query. This observation leads us to an *ellipsis hypothesis*:

> **Ellipsis hypothesis:** For queries formulated into coordinated structures, the subset of catenae that are elliptical candidates identify the salient semantic units in the query.

3.2 Limitations of paths and catenae

The prediction of salient semantic units by catenae is quite robust. However, there are two problems that can limit the effectiveness of any technique that uses catenae or dependency paths in IR.

1) Syntactic ambiguity: We make the simplifying assumption that the most probable parse of a query is accurate and sufficient for the extraction of relevant catenae. However, this is not always true. For example, the sentence '*Is polio under control in China, and _ under observation _?*' constitutes successful ellipsis. The elided words '*polio in china*' are relevant to a base query, '*Is polio under control in China?*'. Unfortunately, in Fig. 1 the elided text does not qualify as a catena. A parse with alternative prepositional phrase attachment is shown in Fig. 4. Here, the successfully elided text does qualify as a catena. This highlights the fact that a single dependency parse may only partially represent the ambiguous semantics of a query. More accurate parsing does not address this problem.

2) Rising: Automatic extraction of catenae is limited by the phenomenon of rising. Let the *governor* of a catena be the word that licenses it (in Fig. 5 '*used*' licenses '*a toxic chemical*' e.g. 'used what?'). Let the *head* of a catena be its parent in a dependency tree. Rising occurs when the head is not the same as the governor. This is frequently seen with *wh*-fronting questions that start *who, what* etc., as well as with many other syntactic discontinuities (Osborne and Groß, 2012). More specifically, rising occurs when a catena is separated from its governor by words that its governor does not dominate, or the catena dominates the governor, as in Fig. 5. Note that in the risen structure, the words for the catena '*chemical as a weapon*' are discontinuous on the surface, interrupted by the word '*used*'.

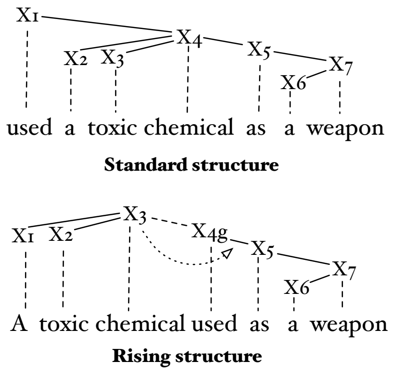


Figure 5: A parse with and without rising. The dashed dependency edge marks where a head is not also the governor and the g-script marks the governor of the risen catena.

4 Selection method for catenae

Catenae describe relatively few of the possible word combinations in a sentence, but still include many combinations that do not result in successful ellipsis and are not informative for IR.

This section describes our supervised method for selection of informative catenae. Candidate catenae are identified using two constraints that enable more efficient extraction: stopwords are removed, and stopped catenae must contain fewer than four words (single words are permitted). We use a pseudo-projective joint dependency parse and semantic role labelling system (Johansson and

Nugues, 2008) to generate the dependency parse. This enables us to explore semantic classification features and is highly accurate. However, any dependency parser may be applied instead. For comparison, catenae extracted from 500 queries using the Stanford dependency parser (de Marneffe et al., 2006) overlap with 77% of catenae extracted from the same queries using the applied parser.

4.1 Feature Classes

Four feature classes are presented in Table 2:

Ellipsis candidates: The ellipsis hypothesis suggests that informative catenae are elliptical candidates. However, queries are not in the coordinated structures required for ellipsis. To enable extraction of characteristic features we (a) construct a *coordinated query* by adding the query to itself; and (b) elide catenae from the second conjunct. For example, for the query, *Is polio under control in China?* we have:

> (a) Is polio under control in China, and is polio under control in China?
> (b) Is polio under control in China, and is polio in China?

We refer to the words in (b) as the *query remainder* and use this to identify features detailed in Table 2.

Dependency path features: Part-of-speech tags and semantic roles have been used to filter dependency paths. We identify several features that use these characteristics from prior work (Table 2).

In addition, variability in the separation distance in documents observed for words that have governor-dependent relations in queries has been proposed for identification of promising paths (Song et al., 2008). We also observe that due to the phenomenon of rising, words that form catenae can be discontinuous in text, and the ability of catenae to match similar word combinations is limited by variability of how they appear in documents. Thus, we propose features for separation distance, but use efficient collection statistics rather than summing statistics for every document in a collection.

Co-occurrence features: A governor w_1 tends to subcategorize for its dependents w_n. This means that w_1 often determines the choice of w_n. We conclude that co-occurrence is an important feature of dependency relations (Mel'čuk, 2003). In addition, term frequencies and inverse document frequencies calculated using word co-occurrence measures are commonly used in IR. We use features previously proposed for filtering terms in IR (Bendersky and Croft, 2008) with two methods to normalize co-occurrence counts for catenae of different lengths: a factor $|c|^{|c|}$, where $|c|$ is the number of words in catena c (Hagen et al., 2011), and the average score for a feature type over all pairwise word combinations in c.

IR performance predictors: Catenae take the same form as typical IR search terms. For this reason, we also use predictors of IR effectiveness previously applied to IR terms.

In general, path and co-occurrence features are similar to those applied by Surdeanu et al. (2011) but we do not parse documents. Path features are also similar to Song et al. (2008), but more efficient and suited to units of variable length. Ellipsis features have not been used before.

5 Experimental setup

5.1 Classification

Catenae selection is framed as a supervised classification problem trained on binary human judgments of *informativeness*: how well catenae represent a query and discriminate between relevant and non-relevant documents in a collection. Kappa for two annotators on catenae in 100 sample queries was 0.63, and test-retest reliability for individual judges was similar (0.62)[3]. Although this is low, human annotations produced consistently better classification accuracy than other labelling methods explored.

We use the Weka (Hall et al., 2009) AdaBoost.M1 meta-classifier (Freund and Schapire, 1996) with unpruned C4.5 decision trees as base learners to classify catenae as informative or not. Adaboost.M1 boosts decisions over T weak learners for T features using weighted majority voting. At each round, predictions of a new learner are focused on incorrectly classified examples from the previous round. Adaboost.M1 was selected in preference to other algorithms because it performed better in preliminary experiments, leverages many weak features to advantage, and usually does not overfit (Schapire et al., 1997).

Predictions are made using 10-fold cross-validation. There are roughly three times the number of uninformative catenae compared to informative catenae. In addition, the number of training examples is small (1295 to 5163 per collection). To improve classifier accuracy, the training data for each collection is supplemented and balanced by generating examples from queries for

[3] Catenae, judgments and annotation details available at `ciir.cs.umass.edu/~tmaxwell`

Ellipsis candidate features (E)	
R_ppl1	Minimum perplexity of ngrams with length 2, 3, and 4 in a window of up to a 3 words around the site of catenae omission. This is the area where ungrammaticality may be introduced. For the remainder R=`ABCDE&ABE' we compute ppl1 for {&ABE, &AB, ABE, &A, AB, BE}.
R_strict	Compliance with strict hand-coded rules for grammaticality of a remainder. Rules include unlikely orderings of punctuation and part-of-speech (POS) tags (e.g. „), poor placement of determiners and punctuation, and orphaned words, such as adjectives without the nouns they modify.
R_relax	A relaxed version of hand-coded rules for *R_strict*. Some rules were observed to be overly aggressive in detection of ungrammatical remainders.
NP_split	Unsuccessful ellipsis often results if elided words only partly describe a base NP. Boolean feature for presence of a partial NP in the remainder. NPs (and PPs) are identified using the MontyLingua toolkit.
PP_split	As for *NP_split*, defined for prepositional phrases (PP).
F_split	As for *NP_split*, defined for finite clauses.
Dependency path features (D)	
c_ppl1	Dependency paths traverse nodes including stopwords and may be filtered based on POS tags. We use perplexity for the sequence of POS tags in catenae before removing stopwords. This is computed using a POS language model built on ukWaC parsed wikipedia data (Baroni et al., 2009).
phClass	Phrasal class for a catena, with options *NP*, *VP* and *Other*. A catena has a NP or VP class if it is, or is entirely contained by, an NP or VP (Song et al., 2008).
semRole	Boolean feature indicating whether a catena describes all, or part of, a predicate-argument structure (PAS). Previous work approximated PAS by using paths between head nouns and verbs, and all paths excluding those within base chunks.
nomEnd	Boolean indicating whether the words at each end of the catena are nouns (or the catena is a single noun).
sepMode	Most frequent separation distance of words in catena *c* in the retrieval collection, with possible values *S* = *{1, 2, 3, long}*. 1 means that all words are adjacent, 2 means separation by 0-1 words, and *long* means containment in a window of size $4 * \lvert c \rvert$.
H_c	Entropy for separation distance *s* of words in catena *c* in the retrieval collection. f_s is the frequency of *c* in window size *s*, and f_S is the frequency of c in a window of size $4 * \lvert c \rvert$. All *f* are normalized for catena length using $\lvert c \rvert^{\lvert c \rvert}$ (Hagen et al., 2011). $H_c = \sum_{s \in S} \frac{f_s + 0.5}{f_S + 0.5} \, log_2 \, \frac{f_s + 0.5}{f_S + 0.5}$
Dependency path features (D) (continued)	
sepRatio	Where f_s and f_S are defined as for *H_c*: $sepRatio_c = \frac{f_{s>2} + 0.5}{f_S + 0.5}$
wRatio	For words *w* in catena *c;* f_S is defined as for *H_c*. $wRatio_c = \frac{0.5 + \frac{1}{\lvert c \rvert} \sum_{w \in c} f_w}{f_S + 0.5}$
Co-occurrence features (C)	
isSeq	Boolean indicating if catena words are sequential in stoplisted surface text.
cf_ow	Frequency of a catena in the retrieval collection, words appearing ordered in a window the length of the catena.
cf_uw	As for *cf_ow*, but words may appear unordered.
cf_uw8	As for *cf_uw*, but the window has a length of 8 words.
idf_ow	Inverse document frequency (*idf*) where document frequency (*df*) of a catena is calculated using *cf_ow* windows. Let N be the number of documents in the retrieval collection, then: $idf(C_i) = log_2 \frac{N}{df(C_i)}$ and $idf(C_i) = N$ if $df(C_i) = 0$.
idf_uw	As for *idf_ow*, but words may appear unordered.
idf_uw8	As for *idf_uw*, but the window has a length of 8 words.
gf	Google ngrams frequency (Brants and Franz, 2006) from a web crawl of approximately one trillion English word tokens. Counts from a large collection are expected to be more reliable than those from smaller test collections.
qf_in	Frequency of appearance in queries from the Live Search 2006 search query log (approximately 15 million queries). Query log frequencies are a measure of the likelihood that a catena will appear in any query.
wf_in	As for *qf_in*, but using frequency counts in Wikipedia titles instead of queries.
IR performance prediction features (I)	
c_len	Length of a stopped catenae. Longer terms tend to reduce IR recall.
WIG	Normalized Weighted Information Gain (*WIG*) is the change in information over top ranked documents between a random ranked list and an actual ranked list retrieved with a catena *c* (Zhou and Croft, 2007). $wig(c) = \frac{\frac{1}{k} \sum_{d \in D_k(c)} log \, p(c \mid d) - log \, p(c \mid C)}{-log \, p(c \mid C)}$ where D_k are the top *k=50* documents retrieved with catena *c* from collection *C*, and $p(c \mid \cdot)$ are maximum likelihood estimates. A second feature uses the average WIG score for all pairwise word combinations in *c*.

Table 2: Classifier features.

	Feature Classes							
	E-D-CI		E-D		E-CI		D-CI	
	Pr	R	Pr	R	Pr	R	Pr	R
ROB04	86.2	72.8	83.5	67.5	86.2	71.7	86.2	72.0
WT10G	79.3	67.1	76.9	59.7	77.2	65.6	79.6	66.1
GOV2	77.0	68.0	70.9	61.8	72.8	63.9	75.5	67.2

Table 3: Average classifier precision (Pr) and recall (R) over 10 folds. Pr is % positive predictions that are correct. R is % positive labeled instances predicted as positive. A combination of all classes marginally performs best.

other collections used in this paper, plus TREC8-QA. For example, training data for Robust04 includes data from WT10G, GOV2 and TREC8-QA. Any examples that replicate catenae in the test collection are excluded. For Robust04, WT10G and GOV2 respectively, 30%, 82% and 69% of the training data is derived from other collections.

5.2 Classification results

Average classification precision and recall is shown in Table 3. Co-occurrence and IR effectiveness prediction features (CI) was the most influential class, and accounted for 70% of all features in the model. Performance is marginally better using all features (E-D-CI) with a moderate improvement over human agreement on the annotation task. The E-D-CI filter is used in subsequent experiments.

Catenae were predicted for all queries. Predictions were more accurate for Robust04 than the other two collections. One potential explanation is that Robust04 queries are longer on average (up to 32 content words per query, compared to up to 16 words) so they generate a more diverse set of catenae that are more easily distinguished with respect to informativeness. The proportion of training data specific to the retrieval collection may also be a factor. Longer queries produce a greater number of catenae, so less training data from other collections is required.

6 Evaluation framework

6.1 Baseline IR models

Baselines are a unigram query likelihood (QL) model (bag of words) and a highly effective sequential dependence (SD) variant of the Markov random field (MRF) model (Metzler and Croft, 2005). SD uses a linear combination of three cliques of terms, where each clique is prioritized by a weight λ_c. The first clique contains individual words (query likelihood QL), $\lambda_1 = 0.85$. The second clique contains query bigrams that match document bigrams in 2-word ordered windows (‘*#1*’), $\lambda_2 = 0.1$. The third clique uses the same bigrams as clique 2 with an 8-word unordered window (‘*#uw8*’), $\lambda_3 = 0.05$. For example, the query *new york city* in Indri[4] query language is:

#weight(
λ_1 #combine(new york city)
λ_2 #combine(#1(new york) #1(york city))
λ_3 #combine(#uw8(new york) #uw8(york city)))

SD is a competitive baseline in IR (Bendersky and Croft, 2008; Park et al., 2011; Xue et al., 2010). Our reformulated model uses the same query format as SD, but the second and third cliques contain filtered catenae instead of query bigrams. In addition, because catenae may be multi-word units, we adjust the unordered window size to $4 * |c|$. So, if two catenae ‘*york*’ and ‘*new york city*’ are selected, the last clique has the form:

λ_3 #combine(york #uw12(new york city))

This query representation enables word relations to be explicitly indicated while maintaining efficient and flexible matching of catenae in documents. Moreover, it does not use dependency relations between words during retrieval, so there is no need to parse a collection.

6.2 Baseline catenae selection

We explore four filters for catenae. Three are based on previous work and describe heuristic features of promising catenae. The fourth is our novel supervised classifier.

NomEnd: Catenae starting and ending with nouns, or containing only one word that is a noun. Paths between nouns are used by Lin and Pantel (2001).

SemRol: Catenae in which all component words are either predicates or argument heads. This is based on work that uses paths between head nouns and verbs (Shen et al., 2005), semantic roles (Moschitti, 2008), and all dependency paths except those that occur between words in the same base chunk (e.g. noun / verb phrase) (Cui et al., 2005).

GovDep: Cantenae containing words with a governor-dependent relation. Many IR models use this form of path filtering e.g. (Gao et al., 2004; Wang et al., 2007). Relations are ‘collapsed’ by removing stopwords to reduce the distance between content nodes in a dependency graph.

[4] `http://www.lemurproject.org/`

	ROBUST04		WT10G		GOV2	
	MAP	R-Pr	MAP	R-Pr	MAP	R-Pr
QL	25.25	28.69	19.55	22.77	25.77	31.26
SD	26.57†	30.02†	20.63	24.31†	28.00†	33.30†
NomEnd	25.91†	29.35‡	20.81†	24.27†	27.41†	32.94†
GovDep	26.26†	29.63†	21.06	24.23†	27.87†	33.51†
SemRol	25.70†	29.06	19.78	22.93	26.76	32.49†
SFeat	**27.04**†	30.11†	20.84†	24.31†	28.43†	33.84†
SF-12	27.03†	**30.20**†	**21.62**†	**24.81**†	**28.57**†	**34.01**†

Table 4: IR results using filtered catenae consistently improve over non-linguistic methods. Significance($p < .05$) shown compared to QL (†) and SD (‡).

	ROBUST04		WT10G		GOV2	
	MAP	R-Pr	MAP	R-Pr	MAP	R-Pr
SF-12	**27.03**	30.20	**21.62**	**24.81**	28.57	34.01
SF-123	26.83	**30.34**	21.34	24.64	**28.77**	**34.24**
SF-NE	26.51	29.86	21.42	24.55	27.96	33.26
SF-GD	26.22	29.48	20.33	23.72	28.30	33.83
Gold	27.92	31.15	22.56	25.69	29.65	35.08

Table 5: Results with supervised selection of catenae with specified length (SF-12, SF-123) are more effective than combinations of SFeat with heuristic NomEnd (SF-NE) or GovDep (SF-GD).

6.3 Experiments

Experiments compare queries reformulated using catenae selected by baseline filters and our supervised selection method (SFeat) to SD and a bag-of-words model (QL). We also compare IR effectiveness of all catenae filtered using SFeat with approaches that combine SFeat with baseline filters. All models are implemented using the Indri retrieval engine version 4.12.

6.4 Results

Results in Table 4 show significant improvement in mean average precision (MAP) of queries using catenae compared to QL. Consistent improvements over SD are also demonstrated for supervised selection applied to all catenae (SFeat) and catenae with only 1-2 words (SF-12) across all collections (Table 5). Overall, changes are small and fairly robust, with one half to two thirds of all queries showing less than 10% change in MAP.

Unlike sFeat, other filters tend to decrease performance compared to SD. Governor-dependent relations for WT10G are an exception and we speculate that this is due to a negative influence of 3-word catenae for this collection. Manual inspection suggests that WT10G queries are short and have relatively simple syntactic structure (e.g. few PP attachment ambiguities). This means that 3-word catenae (in all models except GovDep) tend to include uninformative words, such as '*reasons*' in '*fasting religious reasons*'. In contrast, 3-word catenae in other collections tend to identify query subconcepts or phrases, such as '*science plants water*'.

Classification results for catenae separated by length, such that the classifier for catenae with a specific length are trained on examples of catenae with the same length, confirm this intuition. The rejection rate for 3-word catenae is twice as high for WT10G as for other collections. It is also more difficult to distinguish informative 3-word catenae compared to catenae with 1-2 words. To assess the impact of classification accuracy on IR effectiveness, Table 5 shows results with oracle knowledge of annotator judgments.

The SF-12 model combines catenae predicted for lengths 1 and 2. Its strong performance across all collections suggests that most of the benefit derived from catenae in IR is found in governor-dependent and single word units, where single words are important (GovDep uses only 2-word catenae). Another major observation (Table 5) is that mixing baseline heuristic filters with a supervised approach is not as successful as supervised selection alone. In particular, performance decreases for filtered governor-dependent pairs. This suggests that some important word relations in GovDep and NomEnd are captured by triangulation.

Finally, we review selected catenae for queries that perform significantly better or worse than SD ($>$ 75% change in MAP). The best IR effectiveness occurs when selected catenae clearly focus on the most important aspect of a query. Poor performance is caused by a lack of focus in a catenae set, even though selected catenae are reasonable, or an emphasis on words that are not central to the query. The latter can occur when words that are not essential to query semantics appear in many catenae due to their position in the dependency graph.

7 Conclusion

We presented a flexible implementation of dependency paths for long queries in ad hoc IR that does not require dependency parsing a collection. Our supervised selection technique for catenae addresses the need to balance a representation of language expressiveness with effective, efficient statistical methods. This is a core challenge in computational linguistics.

It is not possible to directly compare performance of our approach with ad hoc techniques in IR that parse a retrieval collection. However, we note that a recent result using query translation based on dependency paths (Park et al., 2011) reports 14% improvement over query likelihood (QL). Our approach achieves 7% improvement over QL on the same collection. We conclude that catenae do not replace path-based techniques, but may offer some insight into their application, and have particular value when it is not practical to parse target documents to determine text similarity.

Acknowledgments

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

References

James Allan, Margaret E. Connell, W. Bruce Croft, Fang-Fang Feng, David Fisher, and Xiaoyan Li. 2000. INQUERY and TREC-9. In *Proceedings of TREC-9*, pages 551–562.

Michael Bendersky and W. Bruce Croft. 2008. Discovering key concepts in verbose queries. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, pages 491–498, New York, NY, USA. ACM.

Keke Cai, Jiajun Bu, Chun Chen, and Guang Qiu. 2007. A novel dependency language model for information retrieval. *Journal of Zhejiang University SCIENCE A*, 8(6):871–882.

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

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC-2006*.

Abdessamad Echihabi and Daniel Marcu. 2003. A noisy-channel approach to question answering. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 16–23, Stroudsburg, PA, USA. Association for Computational Linguistics.

Yoav Freund and Robert E. Schapire. 1996. Experiments with a new boosting algorithm. In *ICML'96*, pages 148–156.

Jianfeng Gao, Jian-Yun Nie, Guangyuan Wu, and Guihong Cao. 2004. Dependence language model for information retrieval. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '04, pages 170–177, New York, NY, USA. ACM.

Matthias Hagen, Martin Potthast, Benno Stein, and Christof Bräutigam. 2011. Query segmentation revisited. In *Proceedings of the 20th international conference on World wide web*, WWW '11, pages 97–106, New York, NY, USA. ACM.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The weka data mining software: an update. *SIGKDD Explorations Newsletter*, 11:10–18, November.

Michael Heilman and Noah A. Smith. 2010. Tree edit models for recognizing textual entailments, paraphrases, and answers to questions. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 1011–1019, Stroudsburg, PA, USA. Association for Computational Linguistics.

Richard Johansson and Pierre Nugues. 2008. Dependency-based syntactic–semantic analysis with PropBank and NomBank. In *Proceedings of CoNNL 2008*, pages 183–187.

Changki Lee, Gary Geunbae Lee, and Myung-Gil Jang. 2006. Dependency structure language model for information retrieval. In *In ETRI journal*, volume 28, pages 337–346.

Dekang Lin and Patrick Pantel. 2001. DIRT - discovery of inference rules from text. In *Proceedings of ACM Conference on Knowledge Discovery and Data Mining (KDD-01)*, pages 323–328, San Francisco, CA.

Loïc Maisonnasse, Eric Gaussier, and Jean-Pierre Chevallet. 2007. 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*, SIGIR '07, pages 695–696, New York, NY, USA. ACM.

Igor A. Mel'čuk. 2003. Levels of dependency in linguistic description: Concepts and problems. In V. Agel, L. Eichinger, H.-W. Eroms, P. Hellwig, H. J. Herringer, and H. Lobin, editors, *Dependency and Valency. An International Handbook of Contemporary Research*, volume 1, pages 188–229. Walter De Gruyter, Berlin–New York.

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

Alessandro Moschitti. 2008. Kernel methods, syntax and semantics for relational text categorization. In *Proceeding of the 17th ACM conference on Information and knowledge management*, CIKM '08, pages 253–262, New York, NY, USA. ACM.

Joakim Nivre. 2005. Dependency grammar and dependency parsing. Technical report, Växjö University: School of Mathematics and Systems Engineering.

Timothy Osborne and Thomas Groß. 2012. Constructions are catenae: Construction grammar meets dependency grammar. *Cognitive Linguistics*, 23(1):165–216.

Timothy Osborne, Michael Putnam, and Groß. 2012. Catenae: Introducing a novel unit of syntactic analysis. *Syntax*, 15(4):354–396, December.

Jae Hyun Park, W. Bruce Croft, and David A. Smith. 2011. A quasi-synchronous dependence model for information retrieval. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, CIKM '11, pages 17–26, New York, NY, USA. ACM.

Ellen F. Prince. 1986. On the syntactic marking of presupposed open propositions. In *Proceedings of the 22nd Annual Meeting of the Chicago Linguistic Society*, pages 208–222.

V. Punyakanok, D. Roth, and W. Yih. 2004. Mapping dependencies trees: An application to question answering. In *Proceedings of AI and MATH Symposium 2004 (Special session: Intelligent Text Processing)*.

Robert E. Schapire, Yoav Freund, Peter Bartlett, and Wee Sun Lee. 1997. Boosting the margin: A new explanation for the effectiveness of voting methods. In *Proceedings of ICML*, pages 322–330.

Dan Shen, Geert-Jan M. Kruijff, and Dietrich Klakow. 2005. Exploring syntactic relation patterns for question answering. In *Proceedings of the Second international joint conference on Natural Language Processing*, IJCNLP'05, pages 507–518, Berlin, Heidelberg. Springer-Verlag.

Fei Song and W. Bruce Croft. 1999. A general language model for information retrieval. In *Proceedings of the 8th ACM international conference on Information and knowledge management*, CIKM '99, pages 316–321, New York, NY, USA. ACM.

Young-In Song, Kyoung-Soo Han, Sang-Bum Kim, So-Young Park, and Hae-Chang Rim. 2008. A novel retrieval approach reflecting variability of syntactic phrase representation. *Journal of Intelligent Information Systems*, 31(3):265–286, December.

Munirathnam Srikanth and Rohini Srihari. 2002. Biterm language models for document retrieval. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '02, pages 425–426, New York, NY, USA. ACM.

Mark J. Steedman. 1990. Gapping as Constituent Coordination. *Linguistics and Philosophy*, 13(2):207–263, April.

Mihai Surdeanu, Massimiliano Ciaramita, and Hugo Zaragoza. 2011. Learning to rank answers to non-factoid questions from web collections. *Computational Linguistics*, 37(2):351–383, June.

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

Mengqiu Wang, Noah A. Smith, and Teruko Mitamura. 2007. What is the Jeopardy model? a quasi-synchronous grammar for QA. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 22–32, Prague, Czech Republic, June. Association for Computational Linguistics.

Xiaobing Xue, Samuel Huston, and W. Bruce Croft. 2010. Improving verbose queries using subset distribution. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, pages 1059–1068, New York, NY, USA. ACM.