An unsupervised method for the extraction of propositional information from text

Simon Dennis*

Institute of Cognitive Science, University of Colorado, Boulder, CO 80301

Recent developments in question-answering systems have demonstrated that approaches based on propositional analysis of source text, in conjunction with formal inference systems, can produce substantive improvements in performance over surface-form approaches. [Voorhees, E. M. (2002) in *Eleventh Text Retrieval Conference*, eds. Voorhees, E. M. & Buckland, L. P., http://trec.nist.gov/pubs/trec11/t11_proceedings.html]. However, such systems are hampered by the need to create broad-coverage knowledge bases by hand, making them difficult to adapt to new domains and potentially fragile if critical information is omitted. To demonstrate how this problem might be addressed, the Syntagmatic Paradigmatic model, a memory-based account of sentence processing, is used to autonomously extract propositional knowledge from unannotated text. The Syntagmatic Paradigmatic model assumes that people store a large number of sentence instances. When trying to interpret a new sentence, similar sentences are retrieved from memory and aligned with the new sentence by using String Edit Theory. The set of alignments can be considered an extensional interpretation of the sentence. Extracting propositional information in this way not only permits the model to answer questions for which the relevant facts are explicitly stated in the text but also allows the model to take advantage of ''inference by coincidence,'' where implicit inference occurs as an emergent property of the mechanism. To illustrate the potential of this approach, the model is tested for its ability to determine the winners of tennis matches as reported on the Association of Tennis Professionals web site.

The closely related fields of question answering and information extraction aim to search large databases of textual material (textbases) to find specific information required by the user (1, 2). As opposed to information retrieval systems, which attempt to identify relevant documents that discuss the topic of the user's information need, information extraction systems return specific information such as names, dates, or amounts that the user requests. Although information retrieval systems (such as Google and Alta Vista) are now in widespread commercial use, information extraction is a much more difficult task and, with some notable exceptions, most current systems are research prototypes. However, the potential significance of reliable information extraction systems is substantial. In military, scientific, and business intelligence gathering, being able to identify specific entities and resources of relevance across documents is crucial. Furthermore, some current information extraction systems now attempt the even more difficult task of providing summaries of relevant information compiled across a document set.

The majority of current information extraction systems are based on surface analysis of text applied to very large textbases. Whereas the dominant approaches in the late 1980s and early 1990s would attempt deep linguistic analysis, proposition extraction, and reasoning, most current systems look for answer patterns within the raw text and apply simple heuristics to extract relevant information (3). Such approaches have been shown to work well when information is represented redundantly in the textbase and when the type of the answer is unambiguously specified by the question and tends to be unique within a given sentence or sentence fragment. Although these conditions often hold for general knowledge questions of the kind found in the Text REtrieval Conference (TREC) Question Answer track, there are many intelligence applications for which they cannot be guaranteed. Often relevant information will be stated only once or may only be inferred and never stated explicitly. Furthermore, the results of the most recent TREC question–answer competition suggest that deep reasoning systems may now have reached a level of sophistication that allows them to surpass the performance possible using surface-based approaches. In the 2002 TREC competition, the POWER ANSWER system (4), which converts both questions and answers into propositional form and uses an inference engine, achieved a confidence weighted score of 0.856, a substantive improvement over the second placed exactanswer (5), which received a score of 0.691 in the main question-answering task.

A key component in the performance of the POWER ANSWER system is its use of the WORDNET lexical database (6). WORDNET provides a catalog of simple relationships among words, such as synonymy, hypernymy, and part-of relations that POWER ANSWER uses to supplement its inference system. Despite the relatively small number of relations considered and the difficulties in achieving good coverage in a hand-coded resource, the additional background knowledge provided by WORDNET significantly improves the performance of the system. This fact suggests that further gains may be achieved if automated methods for extracting a broader range of propositional information could be used in place of, or in conjunction with, the WORDNET database.

In recent years, there have been a number of attempts to build systems capable of extracting propositional information from sentences (7–9).

For instance, given the sentence:

Sampras outguns Agassi in US Open Final,

these systems might produce an annotation such as:

[$_{\text{Winner}}$ Sampras] outguns [$_{\text{Loser}}$ Agassi][$_{\text{Loc}}$ in US Open Final].

This work has been driven, at least in part, by the availability of semantically labeled corpora such as Penn Treebank (10) and FRAMENET (11). As a consequence, the semantic roles used by the systems are those defined by the corpus annotators. However, deciding on a best set of semantic roles has proven extremely difficult. There are a great many schemes that have been proposed ranging in granularity from very broad, such as the two-macro-role

This paper results from the Arthur M. Sackler Colloquium of the National Academy of Sciences, ''Mapping Knowledge Domains,'' held May 9–11, 2003, at the Arnold and Mabel Beckman Center of the National Academies of Sciences and Engineering in Irvine, CA.

Abbreviations: SP, Syntagmatic Paradigmatic; SET, String Edit Theory; RT, relational trace; EM, expectation maximization; LTM, long-term memory.

*E-mail: dennissj@psych.colorado.edu.

www.pnas.org/cgi/doi/10.1073/pnas.0307758101

proposal of ref. 12, through theories that propose nine or 10 roles, such as ref. 13, to much more specific schemes that contain domain-specific slots, such as ORIG_CITY, DEST_CITY, or DEPART_TIME, that are used in practical dialogue understanding systems (14).

That there is much debate about semantic role sets and that existing systems must commit to a scheme *a priori* are important limitations of existing work and, I will argue, are consequences of a commitment to intentional semantics. In systems that use intentional semantics, the meanings of representations are defined by their intended use and have no inherent substructure.

For instance, the statement "Sampras outguns Agassi" might be represented as:

Sampras: Winner
Agassi: Loser

However, the names of the roles are completely arbitrary and carry representational content only by virtue of the inference system in which they are embedded.

Now contrast the above situation with an alternative extensional representation of "Sampras outguns Agassi," in which roles are defined by enumerating exemplars, as follows:

Sampras: Kuerten, Hewitt
Agassi: Roddick, Costa

The winner role is represented by the distributed pattern of Kuerten and Hewitt, words chosen because they are the names of people who have filled the "X" slot in a sentence like "X outguns Y" within the experience of the system. Similarly, Roddick and Costa are the names of people who have filled the "Y" slot in such a sentence and form a distributed representation of the loser role. Note the issue is not just a matter of distributed vs. symbolic representation. The tensor product representation used in the STAR model (15) of analogical reasoning uses distributed representations of the fillers but assigns a unique rank to each role and so is an intentional scheme. By contrast, the temporal binding mechanism proposed by ref. 16 allows for both distributed filler and role vectors and hence could implement extensional semantics.

The use of extensional semantics of this kind has a number of advantages. First, defining a mapping from raw sentences to extensional-meaning representations is much easier than defining a mapping to intentional representations, because it is now necessary only to align sentence exemplars from a corpus with the target sentence. The difficult task of either defining or inducing semantic categories is avoided.

Second, because the role is now represented by a distributed pattern, it is possible for the one-role vector to simultaneously represent roles at different levels of granularity. The pattern {Kuerten, Hewitt} could be thought of as a protoagent, an agent, a winner, and a winner of a tennis match, simultaneously. The role vectors can be determined from a corpus during processing, and no commitment to an *a priori* level of role description is necessary.

Third, extensional representations carry content by virtue of the other locations in the experience of the system where those symbols have occurred. That is, the systematic history of the comprehender grounds the representation. For instance, we might expect systematic overlap between the winner and person-who-is-wealthy roles, because some subset of {Kuerten, Hewitt} may also have occurred in an utterance such as "X is wealthy." These contingencies occur as a natural consequence of the causality being described by the corpus. We will call this type of implicit inference inference by coincidence and, as we will see in subsequent sections, the performance of the model is in large part due to this emergent property.

In the next section, we give a brief introduction to String Edit Theory (SET), which is used in the model to identify sentences from the corpus suitable for alignment with the current target and to define how these sentences should align. Next, the steps involved in interpreting a sentence in the model will be outlined. Then, the Tennis News domain that was chosen to test the model is described, and the results are presented. Finally, some factors that remain to be addressed are discussed.

Introduction to SET

SET was popularized in a book entitled *Time Warps, String Edits and Macromolecules* (17) and has been developed in both the fields of computer science and molecular biology (18–21). As the name suggests, the purpose of SET is to describe how one string, which could be composed of words, letters, amino acids, etc., can be edited to form a second string. That is, what components must be inserted, deleted, or changed to turn one string into another. In the model, SET is used to decide which sentences from a corpus are most like the target sentence, and which tokens within these sentences should align.

As an example, suppose we are trying to align the sentences "Sampras defeated Agassi" and "Kuerten defeated Roddick." The most obvious alignment is that which maps the two sentences to each other in a one-to-one fashion.

$$\begin{array}{ccc} \text{Sampras} & \text{defeated} & \text{Agassi} \\ | & | & | \\ \text{Kuerten} & \text{defeated} & \text{Roddick} \end{array} \qquad \textbf{[A1]}$$

In this alignment, we have three edit operations. There is a change of "Sampras" for "Kuerten," a match of "defeated," and a change of "Agassi" for "Roddick." In fact, this alignment can also be expressed as a sequence of edit operations,

⟨Sampras, Kuerten⟩
⟨defeated, defeated⟩
⟨Agassi, Roddick⟩

In SET, sentences do not have to be of the same length to be aligned. If we add "Pete" to the first sentence, we can use a delete to describe one way in which the resulting sentences could be aligned.

$$\begin{array}{cccc} \text{Pete} & \text{Sampras} & \text{defeated} & \text{Agassi} \\ | & | & | & | \\ – & \text{Kuerten} & \text{defeated} & \text{Roddick} \end{array} \qquad \textbf{[A2]}$$

The "–" symbol is used to fill the slot left by a deletion (or an insertion) and can be thought of as the empty word. The corresponding edit operation is denoted by ⟨Sampras, –⟩. Although these alignments may be the most obvious ones, there are many other options.

For instance, in aligning "Sampras defeated Agassi" and "Kuerten defeated Roddick," we could start by deleting "Sampras."

$$\begin{array}{cccc} \text{Sampras} & \text{defeated} & \text{Agassi} & – \\ | & | & | & | \\ – & \text{Kuerten} & \text{defeated} & \text{Roddick} \end{array} \qquad \textbf{[A3]}$$

Note that "Roddick" is now inserted at the end of the alignment (denoted ⟨–, Roddick⟩).

Alternatively, we could have deleted "Sampras" and then inserted "Kuerten," to give the following.

$$\begin{array}{cccc} \text{Sampras} & – & \text{defeated} & \text{Agassi} \\ | & | & | & | \\ – & \text{Kuerten} & \text{defeated} & \text{Roddick} \end{array} \qquad \textbf{[A4]}$$

There are a total of 63 ways in which "Sampras defeated Agassi" can be aligned with "Kuerten defeated Roddick," but not all of these alignments are equally likely. Intuitively, alignment **A4** seems better than **A3**, because the word "defeated" is matched. However, this alignment still seems worse than **A1** because it requires "Sampras" to be deleted and "Kuerten" to be inserted. A mechanism that produces alignments of sentences should favor those that have many matches and should penalize those that require many insertions and deletions. To capture these intuitions, edit operations are assigned probabilities. Typically, match probabilities are higher than change probabilities, which are higher than insertion or deletion probabilities. Assuming conditional independence of the edit operations, the probability of an alignment is the multiplication of the probabilities of the edit operations of which it is comprised. Each alignment is an exclusive hypothesis about how the two strings might be aligned, and so the probability that the strings are aligned in one of these ways is the addition of the probabilities of the alignments. Given that there are an exponential number of alignments among strings, one may be concerned that any algorithm based on SET would be infeasible. However, there exist efficient dynamic programming algorithms that have O(nm) time and space complexity, where n and m are the lengths of the two strings (20).

Gap Probabilities

In the molecular biology literature, it is common to assign a lower probability to an initial insertion or deletion (known collectively as indels) and then higher probabilities to subsequent indels in the same block. As a consequence, alignments that involve long sequences of indels are favored over alignments that have many short sequences (22–24). In the context of macromolecule alignment, increasing the probability of block indels is desirable, because a single mutation event can often lead to a block insertion or deletion. An analogous argument is also applicable in the language case, because it is common for entire phrases or clauses to differentiate otherwise structurally similar sentences.

To illustrate the point, consider aligning the sentences "Sampras defeated Agassi" and "Sampras who defeated Roddick defeated Agassi." Two possible alignments are:

$$\begin{array}{cccccc} \text{Sampras} & – & – & – & \text{defeated} & \text{Agassi} \\ | & | & | & | & | & | \\ \text{Sampras} & \text{who} & \text{defeated} & \text{Roddick} & \text{defeated} & \text{Agassi} \end{array} \qquad \textbf{[A5]}$$

and

$$\begin{array}{cccccc} \text{Sampras} & – & \text{defeated} & – & – & \text{Agassi} \\ | & | & | & | & | & | \\ \text{Sampras} & \text{who} & \text{defeated} & \text{Roddick} & \text{defeated} & \text{Agassi.} \end{array} \qquad \textbf{[A6]}$$

Note that these alignments have the same matches and deletions and so will have the same probabilities as calculated above. However, Eq. **A5** should be preferred over Eq. **A6**, because it involves the block deletion of a clause. To capture this property, it is assumed that the probability of an initial indel is lower than the probability of a continuing indel. Now Eq. **A5** will be favored because it involves a single start indel and two subsequent indels, whereas Eq. **A6** has two start indels and one subsequent indel.[†] There exists an algorithm that calculates alignment probabilities under this model that retains O(nm) time and space complexity (22).

We have now completed the overview of SET. In the next section, the way in which the model exploits SET is described.

Description of the Syntagmatic Paradigmatic (SP) Model

In the SP model, sentence processing is characterized as the retrieval of associative constraints from sequential and relational long-term memory (LTM) and the resolution of these constraints in working memory. Sequential LTM contains the sentences from the corpus. Relational LTM contains the extensional representations of the same sentences (see Fig. 1).

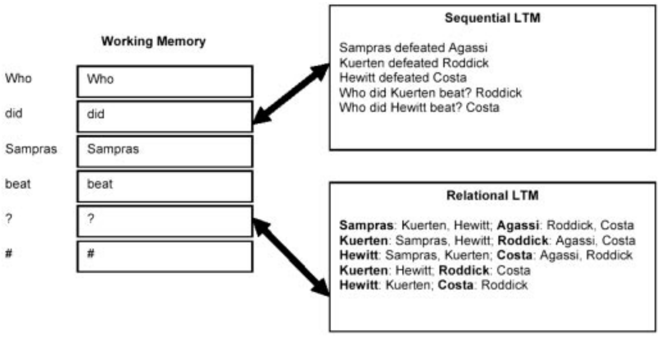
Working Memory
Who
did
Sampras
beat
?
#
Who
did
Sampras
beat
?
#
Sequential LTM
Sampras defeated Agassi
Kuerten defeated Roddick
Hewitt defeated Costa
Who did Kuerten beat? Roddick
Who did Hewitt beat? Costa
Relational LTM
Sampras: Kuerten, Hewitt; **Agassi**: Roddick, Costa
Kuerten: Sampras, Hewitt; **Roddick**: Agassi, Costa
Hewitt: Sampras, Kuerten; **Costa**: Agassi, Roddick
Kuerten: Hewitt; **Roddick**: Costa
Hewitt: Kuerten; **Costa**: Roddick

Fig. 1. SP architecture. #, empty slot. Ultimately, it will contain the answer to the question.

Creating an interpretation of a sentence/utterance involves the following steps.

Sequential Retrieval. The current sequence of input words is used to probe sequential memory for traces containing similar sequences of words. In the example, traces four and five, "Who did Kuerten beat? Roddick," and "Who did Hewitt beat? Costa," are the closest matches to the target sentence "Who did Sampras beat? #" and are assigned high probabilities (see Fig. 2).

To calculate the retrieval strength of a sequential trace, we take a similar approach to that adopted by the Bayesian models of recognition memory (25, 26), which have proven very successful at capturing a variety of memory effects.

Using the terminology $S_k \mapsto T$ to indicate that sequential trace S_k generated the target sentence T, we start with the odds ratio for $S_k \mapsto T$ given T and use the Bayes theorem to convert to a likelihood ratio:

$$\begin{aligned} \frac{P(S_k \mapsto T|T)}{P(\overline{S_k \mapsto T}|T)} &= \frac{P(T|S_k \mapsto T)P(S_k \mapsto T)}{P(T|\overline{S_k \mapsto T})P(\overline{S_k \mapsto T})} \\ &= \frac{P(T|S_k \mapsto T)}{P(T|\overline{S_k \mapsto T})}, \end{aligned} \qquad \textbf{[1]}$$

assuming equal priors.

To calculate $P(T|S_k \mapsto T)$, we use as our data the possible edit sequences that may have been used to transform the trace into the target sentence. Each edit sequence represents an exclusive hypothesis about how S_k generated T, so the probabilities of these hypotheses can be added to determine $P(T|S_k \mapsto T)$:

$$P(T|S_k \mapsto T) = \sum_{a_p \in A_k} P(a_p|S_k \mapsto T), \qquad \textbf{[2]}$$

where a_p is one of the edit sequences relating T and S_k, and A_k is the set of these sequences.

Assuming that the edit operations (match, change, insert, or delete) are sampled independently to create alignments:

$$P(T|S_k \mapsto T) = \sum_{a_p \in A_k} \prod_{e_r \in a_p} P(e_r|S_k \mapsto T), \qquad \textbf{[3]}$$

where e_r is the rth edit operation in alignment a_p.

Similarly,

$$P(T|\overline{S_k \mapsto T}) = \sum_{a_p \in A_k} \prod_{e_r \in a_p} P(e_r|\overline{S_k \mapsto T}). \qquad \textbf{[4]}$$

So, rearranging Eq. **1** and substituting in Eqs. **3** and **4**,

[†] Allison, Wallace, and Yee (21) describe this process in terms of a three-state finite-state machine and also generalize beyond the three-state case. Here, however, only the three-state case will be considered.

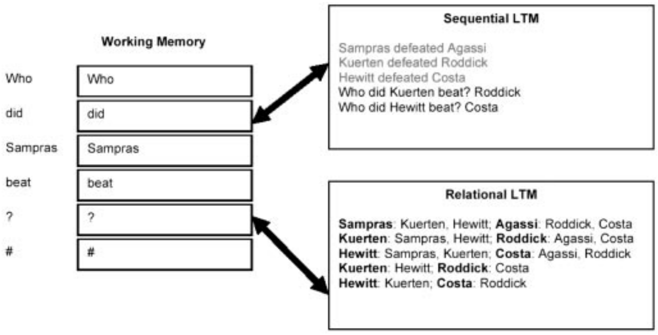
Working Memory
Who
did
Sampras
beat
?
#
Who
did
Sampras
beat
?
#
Sequential LTM
Sampras defeated Agassi
Kuerten defeated Roddick
Hewitt defeated Costa
Who did Kuerten beat? Roddick
Who did Hewitt beat? Costa
Relational LTM
Sampras: Kuerten, Hewitt; Agassi: Roddick, Costa
Kuerten: Sampras, Hewitt; Roddick: Agassi, Costa
Hewitt: Sampras, Kuerten; Costa: Agassi, Roddick
Kuerten: Hewitt; Roddick: Costa
Hewitt: Kuerten; Costa: Roddick

Fig. 2. Sequential retrieval. The traces ''Who did Kuerten beat? Roddick'' and ''Who did Hewitt beat? Costa'' are most similar to the input sentence ''Who did Sampras beat? #'' and are retrieved from sequential LTM. Bold type is used to indicate the traces that are retrieved.

$$P(S_k \mapsto T|T) = \frac{\sum_{a_p \in A_k} \prod_{e_r \in a_p} P(e_r|S_k \mapsto T)}{\sum_{a_p \in A_k} \prod_{e_r \in a_p} P(e_r|S_k \mapsto T) + \sum_{a_p \in A_k} \prod_{e_r \in a_p} P(e_r|\overline{S_k \mapsto T})}. \quad [5]$$

The expected retrieval probability is $P(S_k \mapsto T|T)$ normalized over the traces in memory,

$$\frac{P(S_k \mapsto T|T)}{\sum_j P(S_j \mapsto T|T)}.$$

Sequential Resolution. The retrieved sequences are then aligned with the target sentence to determine the appropriate set of substitutions for each word (see Fig. 3). Note that the slot adjacent to the "#" symbol contains the pattern {Costa, Roddick}. This pattern represents the role that the answer to the question must fill (i.e., the answer is the loser).

During sequential resolution, we are interested in calculating $E_k[P(\langle W_m, T_i\rangle|T)]$, the expected value of the probability that word T_i in slot i in the target sentence substitutes for the word W_m from the lexicon (W) in the context of sentence T.

We can write

$$E_k[P(\langle W_m, T_i\rangle|T)] = \sum_{k=1}^{N} \frac{P(S_k \mapsto T|T)}{\sum_{i=1}^{N} P(S_i \mapsto T|T)} P(\langle W_m, T_i\rangle|S_k \mapsto T, T), \quad [6]$$

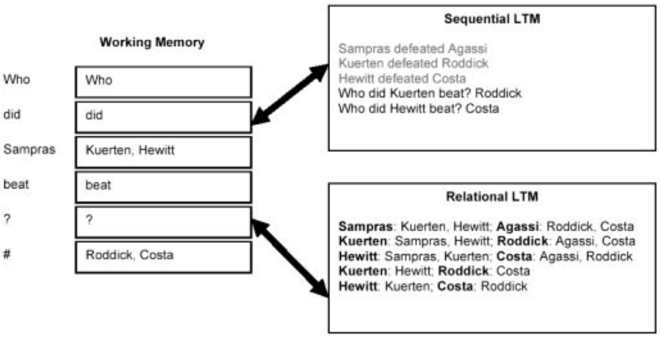
Working Memory
Who
did
Sampras
beat
?
#
Who
did
Kuerten, Hewitt
beat
?
Roddick, Costa
Sequential LTM
Sampras defeated Agassi
Kuerten defeated Roddick
Hewitt defeated Costa
Who did Kuerten beat? Roddick
Who did Hewitt beat? Costa
Relational LTM
Sampras: Kuerten, Hewitt; Agassi: Roddick, Costa
Kuerten: Sampras, Hewitt; Roddick: Agassi, Costa
Hewitt: Sampras, Kuerten; Costa: Agassi, Roddick
Kuerten: Hewitt; Roddick: Costa
Hewitt: Kuerten; Costa: Roddick

Fig. 3. Sequential resolution. Kuerten and Hewitt align with Sampras, and Roddick and Costa align with the answer slot (''#'').

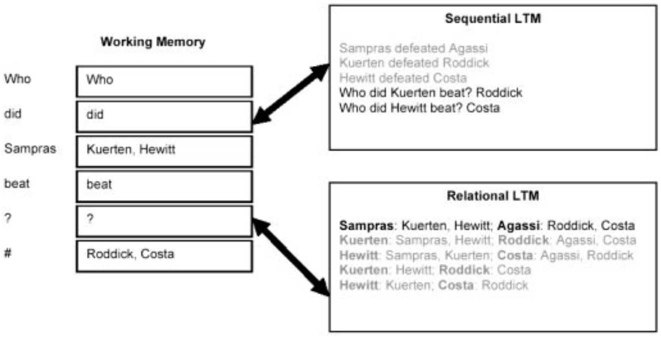
Working Memory
Who
did
Sampras
beat
?
#
Who
did
Kuerten, Hewitt
beat
?
Roddick, Costa
Sequential LTM
Sampras defeated Agassi
Kuerten defeated Roddick
Hewitt defeated Costa
Who did Kuerten beat? Roddick
Who did Hewitt beat? Costa
Relational LTM
Sampras: Kuerten, Hewitt; Agassi: Roddick, Costa
Kuerten: Sampras, Hewitt; Roddick: Agassi, Costa
Hewitt: Sampras, Kuerten; Costa: Agassi, Roddick
Kuerten: Hewitt; Roddick: Costa
Hewitt: Kuerten; Costa: Roddick

Fig. 4. Relational retrieval. The first relational trace is retrieved, because it contains similar role-filler bindings. Bold type is used to indicate the traces that are retrieved.

where N is the number of sequential traces in memory. Now we have divided the task into determining the probability that sequential trace k generated the target (which we calculated in the last section) and determining the probability that T_i and S_{kj} align given that trace k did generate the target.

Calculating the latter is straightforward, now that we have defined how alignments and edit operations are related. Because a given edit operation is either in an alignment or not, we can just add the probabilities of the alignments in which this change occurs and normalize by the probability of all the alignments:

$$\begin{aligned} P(\langle W_m, T_i\rangle|S_k \mapsto T, T) &= \sum_{W_m = S_{kj}} P(\langle S_{kj}, T_i\rangle|S_k \mapsto T, T) \\ &= \sum_{W_m = S_{kj}} \frac{\sum_{\substack{a_p \in A_k \\ \langle S_{kj}, T_i\rangle \in a_p}} P(a_p|S_k \mapsto T)}{\sum_{a_p \in A_k} P(a_p|S_k \mapsto T)}. \end{aligned} \quad [7]$$

We now have an algorithm with which we can calculate the probabilities of substitution within sentential context.

Relational Retrieval. The bindings of input words to their corresponding role vectors (the relational representation of the target sentence) are then used to probe relational LTM. In this case, trace one is favored because it involves similar role-filler bindings. That is, it contains a binding of Sampras onto the {Kuerten, Hewitt} pattern, and it also contains the {Roddick, Costa} pattern. Despite the fact that "Sampras defeated Agassi" has a different surface form than "Who did Sampras beat? #," it contains similar relational information and consequently has a high retrieval probability (Fig. 4).

As in the sequential case, when interpreting a new target sentence, we will assume that its relational trace (RT) has been generated via a set of edit operations on one of the RTs in memory. Specifically, we assume that each binding in RT, which we will denote RT_i, was generated by either an insert or by taking one of the bindings in the RT (R_{kj}) and editing the head word and role vector.

Applying the Bayes rule as we did in the sequential case, we get

$$\begin{aligned} \frac{P(R_k \mapsto RT|RT)}{P(\overline{R_k \mapsto RT}|RT)} &= \frac{P(RT|R_k \mapsto T)P(R_k \mapsto RT)}{P(RT|\overline{R_k \mapsto T})P(\overline{R_k \mapsto RT})}, \\ &= \frac{P(RT|R_k \mapsto RT)}{P(RT|\overline{R_k \mapsto RT})} \end{aligned} \quad [8]$$

assuming equal priors. So we must now calculate $P(RT|R_k \mapsto RT)$. If RT contains M bindings, each of which are generated by independent operations,

$$P(RT|R_k \mapsto RT) = \prod_i^M P(RT_i|R_k \mapsto RT). \qquad \textbf{[9]}$$

Furthermore, each binding in RT was generated from one of the bindings in R_k or by an insert, so

$$P(RT_i|R_k \mapsto RT) = \sum_j^N P(RT_i, R_{kj} \mapsto RT_i|R_k \mapsto RT) + P(RT_i, insert(RT_i)|R_k \mapsto RT), \qquad \textbf{[10]}$$

and

$$\begin{aligned} P(RT|R_k \mapsto RT) &= \prod_i^M P(RT_i|R_k \mapsto RT) \\ &= \prod_i^M \left[\sum_j^N P(RT_i, R_{kj} \mapsto RT_i|R_k \mapsto RT) \right. \\ &\qquad \left. + P(RT_i, insert(RT_i)|R_k \mapsto RT) \right] \\ &= \prod_i^M \left[\sum_j^N P(R_{kj} \mapsto RT_i|R_k \mapsto RT)P \right. \\ &\qquad \left. \cdot (RT_i|R_{kj} \mapsto RT_i) + P(insert(RT_i)|R_k \mapsto RT) \right]. \end{aligned} \qquad \textbf{[11]}$$

Note that if $R_{kj} \mapsto RT_i$, then $R_k \mapsto RT$, so $P(RT_i|R_{kj} \mapsto RT_i, R_k \mapsto RT) = P(RT_i|R_{kj} \mapsto RT_i)$. Also, $P(RT_i, insert(RT_i)|R_k \mapsto RT) = P(insert(RT_i)|R_k \mapsto RT)$. Now $P(RT_i|R_{kj} \mapsto RT_i)$ is the probability that the head word of RT_i (T_i) substitutes for the head word of R_{kj} (S_{kj}), and that the vector of change probabilities of RT_i is an edited version of the R_{kj} vector. To determine the probability of head-word substitution, the prior substitution probability can be used $P(\langle S_{kj}, T_i\rangle|S_k \mapsto T)$. To determine the probability of vector substitution, recall that each of the vectors is comprised of change probabilities. In each case, only one of the words could have substituted for their respective head words, so we can multiply the probability of the trace word (R_{kjl}) by the probability of the target word (RT_{ip}) and the probability that the trace word would substitute for the target word ($P(\langle W_p, W_l\rangle|S_k \mapsto T)$ to obtain an estimate of the probability that the role vector of R_{kj} was edited to produce the role vector of RT_i so

$$P(RT_i|R_{kj} \mapsto RT_i) = P(\langle S_{kj}, T_i\rangle|S_k \mapsto T)\sum_p \sum_l RT_{ip}P(\langle W_p, W_l\rangle|S_k \mapsto T)R_{kjl}, \qquad \textbf{[12]}$$

where RT_{ip} is the pth component of the role vector of RT_i, and R_{kjl} is the lth component of the role vector of R_{kj}. The $P(RT_i|\overline{R_{kj} \mapsto RT_i})$, is calculated in an analogous way by using the $P(\langle S_{kj}, T_i\rangle|\overline{S_k \mapsto T})$ and $P(\langle W_p, W_l\rangle|\overline{S_k \mapsto T})$.

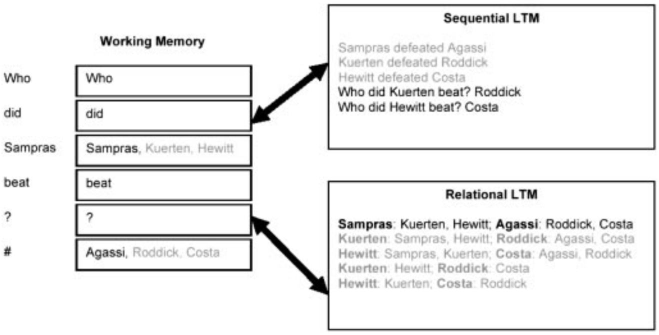
Working Memory
Who
did
Sampras
beat
?
#
Who
did
Sampras, Kuerten, Hewitt
beat
?
Agassi, Roddick, Costa
Sequential LTM
Sampras defeated Agassi
Kuerten defeated Roddick
Hewitt defeated Costa
Who did Kuerten beat? Roddick
Who did Hewitt beat? Costa
Relational LTM
Sampras: Kuerten, Hewitt; Agassi: Roddick, Costa
Kuerten: Sampras, Hewitt; Roddick: Agassi, Costa
Hewitt: Sampras, Kuerten; Costa: Agassi, Roddick
Kuerten: Hewitt; Roddick: Costa
Hewitt: Kuerten; Costa: Roddick

Fig. 5. Relational resolution. Agassi aligns with the answer slot, because it is bound to the {Roddick, Costa} pattern in the retrieved relational trace.

A similar logic is used to calculate the insertion probability

$$P(insert(RT_i)|R_k \mapsto RT) = P(\langle -, T_i\rangle|S_k \mapsto T)\sum_p P(\langle -, T_i\rangle|S_k \mapsto T)RT_{ip}. \qquad \textbf{[13]}$$

And finally the retrieval component is

$$P(R_k \mapsto RT|RT) = \frac{P(RT|R_k \mapsto T)}{P(RT|R_k \mapsto T) + P(RT|\overline{R_k \mapsto T})}. \qquad \textbf{[14]}$$

As before, the expected retrieval probability is $P(R_k \mapsto RT|RT)$ normalized over the traces in memory,

$$\frac{P(R_k \mapsto RT|RT)}{\sum_j P(R_j \mapsto RT|RT)}.$$

The above algorithm has constant space and time complexities $O(|T||S_k||W|^2)$, where $|W|$ is the size of the vocabulary. Although in principle this is expensive, in practice there is typically a small set of traces that attract the majority of the probability mass. Traces with very low retrieval probabilities are truncated and, as a consequence, there are usually only a few nonzero entries in each role vector.

Relational Resolution. Finally, the paradigmatic associations in the retrieved RTs are used to update working memory. In the RT for "Sampras defeated Agassi," "Agassi" is bound to the {Roddick, Costa} pattern. Consequently, there is a strong probability that "Agassi" should align with the "#" symbol, which, as a consequence of sequential retrieval, is also aligned with the {Roddick, Costa} pattern. Note that the model has now answered the question: it was Agassi who was beaten by Sampras (see Fig. 5).

As in the sequential case, we wish to calculate the probability that a given word substitutes for T_i given the relational representation of the target (RT).

$$E_k[P(\langle W_m, T_i\rangle|RT)] = \sum_{k=1}^N \frac{P(R_k \mapsto RT|RT)}{\sum_{i=1}^N P(R_i \mapsto RT|RT)} P(\langle W_m, T_i\rangle|R_k \mapsto RT, RT), \qquad \textbf{[15]}$$

where $P(R_k \mapsto RT|RT)$ was calculated in the last section. To calculate the probability of substitution, we note that a substitution of T_i for S_{kj} has occurred whenever $R_{kj} \mapsto RT_i$. As a consequence, the following derivation applies.

$$P(\langle W_m, T_i\rangle | R_k \mapsto RT, RT)$$
$$= \sum_{W_m = S_{kj}} P(\langle S_{kj}, T_i\rangle | R_k \mapsto RT, RT)$$
$$= \sum_{W_m = S_{kj}} \frac{P(RT_i | R_{kj} \mapsto RT_i)}{\sum_j P(RT_i | R_{kj} \mapsto RT_i) P(R_{kj} \mapsto RT_i | R_k \mapsto RT) + P(insert(RT_i | R_k \mapsto RT)}. \quad [16]$$

Combining Sequential and Relational Substitution Probabilities

We now have two procedures by which we can generate estimates of the substitution probabilities of trace and target words, one based on the sequence of words in the target (sequential) and one based on retrieved role-filler bindings (relational). The final question is how the estimates based on these two different sources of information should be combined to arrive at a final set of substitution probabilities. Taking a simple mixture of the information sources, we get:

$$P(\langle W_m, T_i\rangle) = \eta P(\langle W_m, T_i\rangle | T) + (1 - \eta) P(\langle W_m, T_i\rangle | RT), \quad [17]$$

where η is set at 0.5 for the simulations reported here.[‡]

To summarize, the model hypothesizes four basic steps. First, the series of words in the target sentence is used to retrieve traces that are similar from sequential LTM. Then, the retrieved sequential traces are aligned with the input sentence to create a relational interpretation of the sentence based on word order. This interpretation is then used to retrieve similar traces from relational LTM. Finally, working memory is updated to reflect the relational constraints retrieved in the previous step.

Updating Edit Probabilities with Corpus Statistics

The method used to derive the edit probabilities used above is a version of the Expectation Maximization (EM) algorithm (27). EM algorithms involve two steps. In the first step, the expected value of the log likelihood of the data given the current parameters is calculated. That is, we define Q:

$$Q(\theta, \theta^t) = \int_{y \in Y} \log(P(C, y|\theta) P(y|C, \theta^t) d\theta, \quad [18]$$

where C is the set of sentences in the training corpus, and Y is the set of all possible hidden variables (i.e., trace selections and edit sequences) that could have given rise to the set of traces. θ^t is the set of parameters at the current step.

In the second step, we find the parameters θ, which maximize Q. These will be used as the parameters for the next iteration of the algorithm

$$\theta^{t+1} = \arg\max_{\theta} Q(\theta, \theta^t). \quad [19]$$

Repeated iterations of the EM algorithm are guaranteed to find a local minimum in the log likelihood (27). In the case of the SP model, the training algorithm reduces to adding the probabilities of the alignments in which each edit operation occurs and normalizing appropriately. Space precludes providing the entire derivation, but it follows the familiar pattern of EM derivations of mixture models (28).

Although the EM algorithm has proven useful in a wide range of language-learning tasks, optimization of the log likelihood of the data is not always a desirable objective (29). In the case of the SP model, a difficulty arises with the optimization of match probabilities. For low-frequency words, the probability that there will be a match of these words in the corpus can be very small, meaning that the match probabilities tend to zero. This property is particularly undesirable when the match probabilities are used in the relational model. For that reason, only change and indel probabilities were trained in the following evaluation.

The mathematical framework of the SP model has now been outlined. In the next section, we describe the data set used to test the question-answering capabilities of the model.

The Tennis News Domain

A number of criteria were used to select the domain on which to test the model. First, the domain was required to be one for which naturally occurring text was available, because it is important that the model be capable of dealing robustly with the variety of sentences typically found in real text. Also, in real corpora, there are many sentences that do not refer to the facts of interest at all, and the model should be capable of isolating the relevant ones.

Second, we wished to test the model's ability to extract relational information from sentences. Many question-answering systems use type heuristics rather than engaging in relational analysis. For instance, they might determine the date of the running of the Melbourne Cup by looking for sentences containing the term Melbourne Cup and returning any date within these sentences regardless of the role this date might fill. Although such heuristics are often very successful in practice, there are some questions for which a relational analysis is necessary.

Finally, we were interested in testing the model's ability to take advantage of inference by coincidence and so chose a domain in which the opportunities for such inferences are abundant.

Sixty-nine articles were taken from the Association of Tennis Professionals web site (www.atptennis.com). The articles were written between September 2002 and December 2002 and ranged in length from 134 to 701 words. In total, there are 21,212 words. The documents were manually divided into sentences, and the mean sentence length was 23.7.

The tennis domain fulfills each of the criteria. Naturally occurring text is available, and there were many nontarget sentences that the model was required to reject in its search for relevant information. Choosing the winner of a tennis match cannot be done by appealing to simple type heuristics, because relevant source sentences often contain the names of both the winner and the loser so that the correct answer must be selected from items of the same type. Finally, in sports reporting of this kind, there are often multiple cues, many of which are indirect, that allow the disambiguation of key facts, like who the winner of a match was.

Then 377 questions of the form "Who won the match between X and Y? X" were created. Any result that could be deduced from the article text was included. So, for instance, results that required the resolution of an anaphoric reference from other sentences in the same document were retained. Also, the winner was alternated between the first and second name positions so that the model could not simply repeat the name in the first slot to answer the question.

Results and Discussion

To test the model, the EM algorithm was first used to train edit probabilities and then each question was presented with the final answer slot vacant (e.g., "Who won the match between Sampras and Agassi? #"). The SP model was invoked to complete the pattern.[§]

[‡] I thank an anonymous reviewer who suggested using the explicit mixture model to combine sources.

[§] The parameters used were P(Match) = 0.95 P(Change) = 0.025 P(Not Match) = 0.35, P(Not Change) = 0.45.

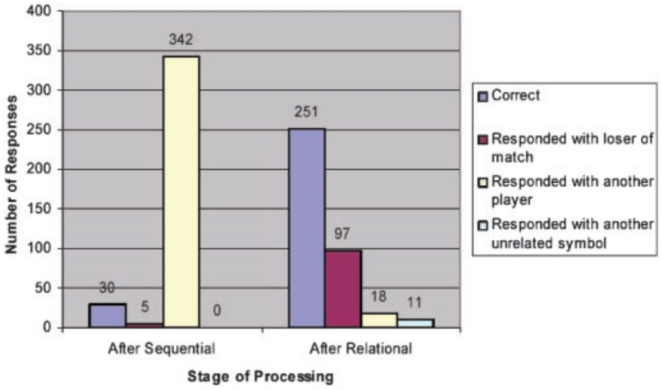

Fig. 6. Breakdown of result types after sequential and relational processing.

During sequential retrieval, the model was allowed to retrieve any sentence or question from the entire corpus. During relational retrieval, however, only facts derived from the sentences were allowed as retrieval candidates,¶ that is, the factual knowledge embodied by the questions was not permitted to influence the results.

The token with the highest probability in the # slot was assumed to be the answer returned by the model. Fig. 6 shows a breakdown of the number of results in each category after sequential and relational resolution. After relational processing, on ≈67% of occasions the model correctly returned the winner of the match. Twenty-six percent of the time, it incorrectly produced the loser of the match. Five percent of the time, it responded with a player other than either the winner or loser of the match, and on 3% of occasions it committed a type error, responding with a word or punctuation symbol that was not a player's name.

There are a number of ways in which one might seek to establish an appropriate baseline against which to compare these results. Because the model is developed in a pattern-completion framework, it is possible for any symbol in the vocabulary to be returned. There were 2,522 distinct tokens in the corpus, so nominally the chance rate is <1%. However, one might also argue that the chance rate should be related to the number of elements of the appropriate type for a response, that is the number of names of players. There were 142 distinct players' names and so, by this analysis, the baseline would also be <1%. A further type distinction would be between winners and losers. There were 85 distinct winners, which results in a baseline of just >1%. Note that in any of these cases, the model is performing well above chance.

Another possible model for the decision process against which one might be tempted to compare the performance of the SP model is a winner maximum-likelihood model. In this model, the two players are extracted from the question, and the one that most often fills the winner slot is selected. With this model, performance is 74%. However, it is important to recognize that, to apply this model, one must provide a mechanism by which the relevant contenders are extracted from the sentence and which is capable of deciding what statistics are relevant for making frequency comparisons, decisions that will change on a question-by-question basis. By contrast, the SP model is only given a pattern to complete, and so is not only answering the question but is also extracting the relevant schema within which the question must be answered. In addition, when the SP model is run without relational retrieval or resolution, performance drops from 67% to 8% correct (see Fig. 6), so it would seem that relational processing was critical. Given that the questions were not included in relational memory, performance must have been driven by the statistics of the articles rather than of the questions. Consequently, the comparison against the maximum-likelihood model is somewhat inappropriate.

Issues That Compromised Performance

In examining the types of errors committed by the model, a number of recurring types were evidenced. As mentioned earlier, the use of anaphora is quite common in this corpus. The current model has no mechanism for the resolution of anaphora, which undermines its ability to both isolate the sentences containing the appropriate relational information and select the correct answer token. In addition, a mechanism for isolating appropriate context is necessary. On seven occasions in the current data set, there are sets of players for whom the questions are ambiguous without the use of context to isolate the correct match. In addition, inference by coincidence can sometimes induce an incorrect response. For instance, the model induces that Schalken won the match against Pete Sampras in part on the basis of the sentence "Schalken, from the Netherlands, made his best-ever grand slam showing at the US open last month. . ." However, although having a best-ever showing is indicative of winning, in this case, it is misleading because it was in fact Sampras who defeated Schalken in the semifinals. Finally, the model's lack of sensitivity to sublexical structure creates difficulties, particularly in deriving relational match when possessives are used. There are then many avenues by which one could look to improve performance.

Inference by Coincidence

To assess the contribution that inference by coincidence made to the performance of the model, the sentence with maximal retrieval probability for each query was classified into one of three categories.

The literal category contained those sentences where there was an explicit statement of the result, even if it required some interpretation. For example, when processing the question, "Who won the match between Ulihrach and Vicente? Ulihrach," the highest-probability relational trace was "Vicente bounced by Ulihrach," which literally states the result (even if it is necessary for one to interpret "bounced" in this context).

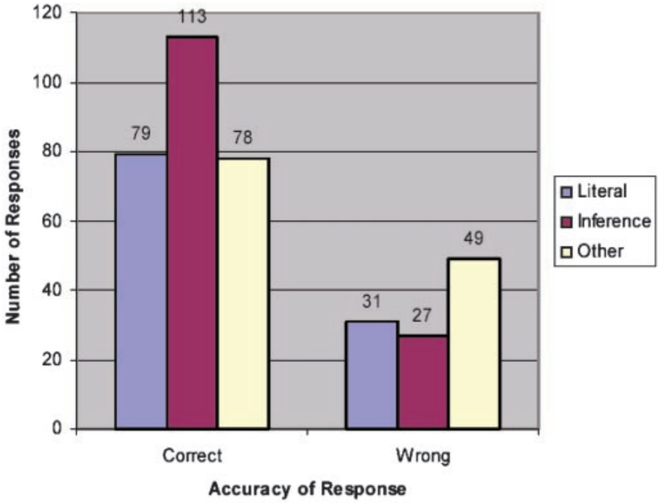

Fig. 7. Breakdown of responses based on the accuracy of the response and the type of the most probable relational trace according to the model. ''Literal'' refers to traces in which the answer was stated explicitly. ''Inference'' refers to traces in which the answer was not stated, but from which it could be inferred. ''Other'' refers to traces from which the answer was not derivable. Note that these statistics are, for the most part, probably trace only. The model, however, accumulates information from multiple traces, so it is still possible for it to answer correctly even if the most probable trace does not contain the relevant information.

¶To speed computation, only sentences that contained at least one of the two combatants were considered.

Table 1. Examples of inference by coincidence in the Tennis News domain

Who won the match between Carlsen and Kiefer? Carlsen.	Kafelnikov now meets Kenneth Carlsen of Denmark in the second round.
Who won the match between Kiefer and Safin? Safin.	Safin, Kafelnikov surge toward hometown showdown.
Who won the match between Ljubicic and Kutsenko? Ljubicic.	Sixth seed Davide Sanguinetti of Italy and eighth seed Ivan Ljubicic of Croatia took different paths to their opening-round wins at the president's cup in Tashkent.
Who won the match between Voltchkov and Haas? Voltchkov.	According to Haas, the injury first arose during Wednesday's match against Sargis Sargsian, and became progressively worse during practice and then the match against Voltchkov.
Who won the match between Srichaphan and Lapentti? Srichaphan.	Srichaphan has now won two titles in four finals this year.
Who won the match between Mamiit and Coria? Coria.	Kuerten, Coria withstand heat, set up fiery South American showdown.

Each example shows the question and the sentence that generated the most probable relational trace.

The inference category included those sentences that did not contain a literal statement of the result but that provided some evidence (not necessarily conclusive) for what the result may have been (see Table 1 for examples). For instance, when processing the question, "Who won the match between Portas and Sampras? Sampras," the relational trace with the highest retrieval probability was "Sampras claims 14th Grand Slam title." Although this sentence does not explicitly state the result of this match, one can infer that if Sampras won the title, then it is likely that he won this match. Note that this inference does not always follow, because the writer may have made reference to a result from a different tournament, or the question may have come from a different article. However, that Sampras won the title does provide evidence in favor of his having won this match. Unlike a traditional inference system, however, the SP model is making the inference by virtue of the fact that the names of people that appear in statements of the form "X claims title" also tend to appear in the winner slot at the end of the questions.

Finally, the other category included all remaining cases. These contained traces in which both players were mentioned but the sentence could not have been used to conclude who the winner may have been. For example, when the question, "Who won the match between Acasuso and Pavel? Acasuso" was presented, the most probable relational trace was "Pavel and Acasuso to clash in Bucharest semis." In addition, this category contains sentences that contradict the correct result. For example, the question "Who won the match between Pavel and Srichaphan? Pavel" produced the relational trace "Pavel, now 38-22 on the year, has reached two semifinals in 2002 Chennai I. to Srichaphan and Bucharest I. To Acasuso." This situation occurs when a player revenges an earlier loss. In addition, the other category was assigned when the sentence was unrelated to the question. For instance, when the model was presented with the question, "Who won the match between Meligeni and Etlis? Etlis," it returned "Kiefer quickly overcame Gaston Etlis of Argentina 6-2, 6-4 on Monday to qualify for the main draw of the Kremlin cup."

Fig. 7 shows the number of most probable relational traces in each category.

For an indication of the contribution that inference by coincidence is making to correct responding, consider those correct responses that can be attributed to either literal or inference traces. On 59% of occasions, the model was inferring the answer rather than relying on literal retrieval. Given that in each case a literal statement of the results existed in the corpus, it is significant that inference by coincidence seems to be playing such a crucial role in the performance of the model.

Conclusion

The ability of the SP model to isolate the combatants from arbitrary sentences and to successfully separate winners from losers demonstrates it is capable of extracting propositional information from text. Using simple retrieval and alignment operations, the model takes advantage of the statistics of word use. Unlike existing work (7, 8, 10), it need make no *a priori* commitment to particular grammars, heuristics, or sets of semantic roles, and it does not require an annotated corpus on which to train.

Furthermore, the large number of occasions (59%) on which the most probable relational trace was a sentence from which the result could be inferred but not directly derived is an indication that inference by coincidence can play a dominant role in successful question answering and may be a crucial factor in sentence comprehension in general.

I acknowledge the many discussions that have shaped the current work. In particular, I thank Michael Harrington, Michael Humphreys, Peter Kwantes, Andrew Smith, Walter Kintsch, Tom Landauer, Jose Quesada, and Michael Littman for helpful comments and suggestions. I also thank Jose Quesada for his assistance in creating the Tennis News data set. This research was supported by Australian Research Council Grant A00106012, U.S. National Science Foundation Grant EIA-0121201, and U.S. Department of Education Grant R305G020027.

1. Gaizauskas, R. & Wilks, Y. (1998) *J. Doc.* **54,** 70–105.
2. Cowie, J. & Lehnert, W. (1996) *Commun. ACM* **39,** 80–91.
3. Voorhees, E. M. (2002) *Eleventh Text Retrieval Conference (TREC 2002)*, eds. Voorhees, E. M. & Buckland, L. P., http://trec.nist.gov/pubs/trec11/t11_proceedings.html.
4. Moldovan, D., Harabagiu, S., Girju, R., Morarescu, P., Lacatusu, F., Novischi, A., Badulescu, A. & Bolohan, O. (2002) *Eleventh Text Retrieval Conference (TREC 2002)*, eds. Voorhees, E. M. & Buckland, L. P., http://trec.nist.gov/pubs/trec11/t11_proceedings.html.
5. Soubbotin, M. M. & Soubbotin, S. M. (2002) in *Eleventh Text Retrieval Conference (TREC 2002)*, eds. Voorhees, E. M. & Buckland, L. P., http://trec.nist.gov/pubs/trec11/t11_proceedings.html.
6. Fellbaum, C. (1998) WORDNET, *An Electronic Lexical Database* (MIT Press, Cambridge, MA).
7. Blaheta, D. & Charniak, E. (2000) in *Proceedings of the 1st Annual Meeting of the North American Chapter of the Association for Computational Linguistics* (North Am. Chapter for the Assoc. for Computational Linguistics, Seattle), pp. 234–240.
8. Gildea, D. & Jurafsky, D. (2002) *Comput. Ling.* **28,** 245–288.
9. Palmer, M., Rosenzweig, J. & Cotton, S. (2001) *Proceedings of the First International Conference on Human Language Technology Research*, ed. Allan, J. (Morgan Kaufmann, San Francisco).
10. Marcus, M., Kim, G., Marcinkiewicz, M. A., MacIntyre, R., Bies, A., Ferguson, M., Katz, K. & Schasberger, B. (1993) *Comput. Ling.* **19,** 313–330.
11. Fillmore, C. J., Wooters, C. & Baker, C. F. (2001) in *Proceedings of the Pacific Asian Conference on Language, Information and Computation* (Pacific Asian Conference on Language, Information and Computation, Hong Kong).
12. Van Valin, R. D. (1993) *Advances in Role and Reference Grammar*, ed. Van Valin, R. D. (John Benjamins, Amsterdam).
13. Fillmore, C. J. (1971) in *22nd Round Table, Linguistics: Developments of the Sixties—Viewpoints of the Seventies*, ed. O'Brien, R. J. (Georgetown Univ. Press, Washington, DC), Vol. 24, pp. 35–56.
14. Stallard, D. (2000) in *Proceedings of the 6th Applied Natural Language Processing Conference (ANLP'00)*, (citeseer.nj.nec.com/stallard00talkntravel.html), pp. 68–75.
15. Halford, G., Wilson, W., Guo, K., Gayler, R., Wiles, J. & Stewart, J. (1994) in *Analogical Connections,* eds. Holyoak, K. J. & Barnden, J. (Ablex, Norwood, MN), Vol. 2, pp. 363–415.
16. Hummel, J. & Holyoak, K. J. (1997) *Psychol. Rev.* **104,** 427–466.
17. Sankoff, D. & Kruskal, J. B. (1983) *Time Warps, String Edits and Macromolecules: The Theory and Practise of Sequence Comparison* (Addison–Wesley, New York).
18. Sellers, P. H. (1974) *J. Combin. Theor.* **16,** 253–258.
19. Levenshtein, V. I. (1965) *Dokl. Akad. Nauk. SSSR* **163,** 845–848.
20. Needleman, S. B. & Wunsch, C. D. (1970) *J. Mol. Biol.* **48,** 443–453.
21. Allison, L., Wallace, C. S. & Yee, C. N. (1992) *J. Mol. Evol.* **35,** 77–89.
22. Gotoh, O. (1982) *J. Mol. Biol.* **162,** 705–708.
23. Waterman, M. S., Smith, T. F. & Beyer, W. A. (1976) *Adv. Math.* **20,** 367–387.
24. Waterman, M. S. (1984) *Bull. Math. Biol.* **46,** 473–500.
25. Shiffrin, R. M. & Steyvers, M. (1997) *Psychon. Rev.* **4,** 145–166.
26. Dennis, S. & Humphreys, M. S. (2001) *Psychol. Rev.* **108,** 452–478.
27. Dempster, A. P., Laird, N. M. & Rubin, D. B. (1977) *J. R. Stat. Soc. B* **39,** 1–38.
28. Bilmes, J. A. (1998) Ph.D. thesis (University of California, Berkeley).
29. Klein, D. & Manning, C. D. (2001) in *Proceedings of the Conference on Computational Natural Language Learning*, eds. Daclemans, W. & Zajac, R. (Toulouse, France), pp. 113–120.