Using Semantic Roles to Improve Question Answering

Dan Shen
Spoken Language Systems
Saarland University
Saarbruecken, Germany
dan@lsv.uni-saarland.de

Mirella Lapata
School of Informatics
University of Edinburgh
Edinburgh, UK
mlap@inf.ed.ac.uk

Abstract

Shallow semantic parsing, the automatic identification and labeling of sentential constituents, has recently received much attention. Our work examines whether semantic role information is beneficial to question answering. We introduce a general framework for answer extraction which exploits semantic role annotations in the FrameNet paradigm. We view semantic role assignment as an optimization problem in a bipartite graph and answer extraction as an instance of graph matching. Experimental results on the TREC datasets demonstrate improvements over state-of-the-art models.

1 Introduction

Recent years have witnessed significant progress in developing methods for the automatic identification and labeling of semantic roles conveyed by sentential constituents.[1] The success of these methods, often referred to collectively as *shallow semantic parsing* (Gildea and Jurafsky, 2002), is largely due to the availability of resources like FrameNet (Fillmore et al., 2003) and PropBank (Palmer et al., 2005), which document the surface realization of semantic roles in real world corpora.

More concretely, in the FrameNet paradigm, the meaning of predicates (usually verbs, nouns, or adjectives) is conveyed by *frames*, schematic representations of situations. Semantic roles (or *frame elements*) are defined for each frame and correspond to salient entities present in the evoked situation. Predicates with similar semantics instantiate the same frame and are attested with the same roles. The FrameNet database lists the surface syntactic realizations of semantic roles, and provides annotated example sentences from the British National Corpus. For example, the frame *Commerce_Sell* has three core semantic roles, namely *Buyer*, *Goods*, and *Seller* — each expressed by an indirect object, a direct object, and a subject (see sentences (1a)–(1c)). It can also be attested with non-core (peripheral) roles (e.g., *Means, Manner*, see (1d) and (1e)) that are more generic and can be instantiated in several frames, besides *Commerce_Sell*. The verbs *sell*, *vend*, and *retail* can evoke this frame, but also the nouns *sale* and *vendor*.

(1) a. [Lee]$_{Seller}$ sold a textbook [to Abby]$_{Buyer}$.
b. [Kim]$_{Seller}$ sold [the sweater]$_{Goods}$.
c. [My company]$_{Seller}$ has sold [more than three million copies]$_{Goods}$.
d. [Abby]$_{Seller}$ sold [the car]$_{Goods}$ [for cash]$_{Means}$.
e. [He]$_{Seller}$ [reluctanctly]$_{Manner}$ sold [his rock]$_{Goods}$.

By abstracting over surface syntactic configurations, semantic roles offer an important first step towards deeper text understanding and hold promise for a range of applications requiring broad coverage semantic processing. Question answering (QA) is often cited as an obvious beneficiary of semantic

[1] The approaches are too numerous to list; we refer the interested reader to Carreras and Màrquez (2005) for an overview.

Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 12–21, Prague, June 2007. ©2007 Association for Computational Linguistics

role labeling (Gildea and Jurafsky, 2002; Palmer et al., 2005; Narayanan and Harabagiu, 2004). Faced with the question *Q: What year did the U.S. buy Alaska?* and the retrieved sentence *S:...before Russia sold Alaska to the United States in 1867*, a hypothetical QA system must identify that *United States* is the *Buyer* despite the fact that it is attested in one instance as a subject and in another as an object. Once this information is known, isolating the correct answer (i.e., *1867*) can be relatively straightforward.

Although conventional wisdom has it that semantic role labeling ought to improve answer extraction, surprising little work has been done to this effect (see Section 2 for details) and initial results have been mostly inconclusive or negative (Sun et al., 2005; Kaisser, 2006). There are at least two good reasons for these findings. First, shallow semantic parsers trained on declarative sentences will typically have poor performance on questions and generally on out-of-domain data. Second, existing resources do not have exhaustive coverage and recall will be compromised, especially if the question answering system is expected to retrieve answers from unrestricted text. Since FrameNet is still under development, its coverage tends to be more of a problem in comparison to other semantic role resources such as PropBank.

In this paper we propose an answer extraction model which effectively incorporates FrameNet-style semantic role information. We present an automatic method for semantic role assignment which is conceptually simple and does not require extensive feature engineering. A key feature of our approach is the comparison of dependency relation paths attested in the FrameNet annotations and raw text. We formalize the search for an optimal role assignment as an optimization problem in a bipartite graph. This formalization allows us to find an exact, globally optimal solution. The graph-theoretic framework goes some way towards addressing coverage problems related with FrameNet and allows us to formulate answer extraction as a graph matching problem. As a byproduct of our main investigation we also examine the issue of FrameNet coverage and show how much it impacts performance in a TREC-style question answering setting.

In the following section we provide an overview of existing work on question answering systems that exploit semantic role-based lexical resources. Then we define our learning task and introduce our approach to semantic role assignment and answer extraction in the context of QA. Next, we present our experimental framework and data. We conclude the paper by presenting and discussing our results.

2 Related Work

Question answering systems have traditionally depended on a variety of lexical resources to bridge surface differences between questions and potential answers. WordNet (Fellbaum, 1998) is perhaps the most popular resource and has been employed in a variety of QA-related tasks ranging from query expansion, to axiom-based reasoning (Moldovan et al., 2003), passage scoring (Paranjpe et al., 2003), and answer filtering (Leidner et al., 2004). Besides WordNet, recent QA systems increasingly rely on syntactic information as a means of abstracting over word order differences and structural alternations (e.g., passive vs. active voice). Most syntax-based QA systems (Wu et al., 2005) incorporate some means of comparison between the tree representing the question with the subtree surrounding the answer candidate. The assumption here is that appropriate answers are more likely to have syntactic relations in common with their corresponding question. Syntactic structure matching has been applied to passage retrieval (Cui et al., 2005) and answer extraction (Shen and Klakow, 2006).

Narayanan and Harabagiu (2004) were the first to stress the importance of semantic roles in answering complex questions. Their system identifies predicate argument structures by merging semantic role information from PropBank and FrameNet. Expected answers are extracted by performing probabilistic inference over the predicate argument structures in conjunction with a domain specific topic model. Sun et al. (2005) incorporate semantic analysis in their TREC05 QA system. They use ASSERT (Pradhan et al., 2004), a publicly available shallow semantic parser trained on PropBank, to generate predicate-argument structures which subsequently form the basis of comparison between question and answer sentences. They find that semantic analysis does not boost performance due to the low recall of the semantic parser. Kaisser (2006) proposes a

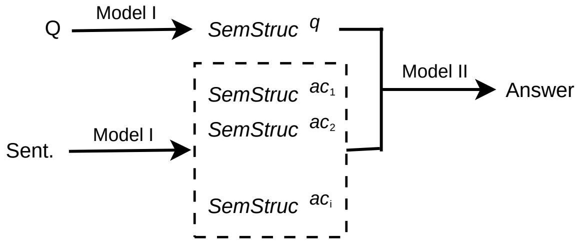

Figure 1: Architecture of answer extraction

question paraphrasing method based on FrameNet. Questions are assigned semantic roles by matching their dependency relations with those attested in the FrameNet annotations. The assignments are used to create question reformulations which are submitted to Google for answer extraction. The semantic role assignment module is not probabilistic, it relies on strict matching, and runs into severe coverage problems.

In line with previous work, our method exploits syntactic information in the form of dependency relation paths together with FrameNet-like semantic roles to smooth lexical and syntactic divergences between question and answer sentences. Our approach is less domain dependent and resource intensive than Narayanan and Harabagiu (2004), it solely employs a dependency parser and the FrameNet database. In contrast to Kaisser (2006), we model the semantic role assignment and answer extraction tasks numerically, thereby alleviating the coverage problems encountered previously.

3 Problem Formulation

We briefly summarize the architecture of the QA system we are working with before formalizing the mechanics of our FrameNet-based answer extraction module. In common with previous work, our overall approach consists of three stages: (a) determining the expected answer type of the question, (b) retrieving passages likely to contain answers to the question, and (c) performing a match between the question words and retrieved passages in order to extract the answer. In this paper we focus on the last stage: question and answer sentences are normalized to a FrameNet-style representation and answers are retrieved by selecting the candidate whose semantic structure is most similar to the question.

The architecture of our answer extraction module is shown in Figure 1. Semantic structures for questions and sentences are automatically derived using the model described in Section 4 (Model I). A semantic structure $SemStruc = \langle p, Set(SRA) \rangle$ consists of a predicate p and a set of semantic role assignments $Set(SRA)$. p is a word or phrase evoking a frame F of FrameNet. A semantic role assignment SRA is a ternary structure $\langle w, SR, s \rangle$, consisting of frame element w, its semantic role SR, and score s indicating to what degree SR qualifies as a label for w.

For a question q, we generate a semantic structure $SemStruc^q$. Question words, such as *what*, *who*, *when*, etc., are considered expected answer phrases (*EAP*s). We require that *EAP*s are frame elements of $SemStruc^q$. Likely answer candidates are extracted from answer sentences following some preprocessing steps detailed in Section 6. For each candidate ac, we derive its semantic structure $SemStruc^{ac}$ and assume that ac is a frame element of $SemStruc^{ac}$. Question and answer semantic structures are compared using a model based on graph matching detailed in Section 5 (Model II). We calculate the similarity of all derived pairs $\langle SemStruc^q, SemStruc^{ac} \rangle$ and select the candidate with the highest value as an answer for the question.

4 Semantic Structure Generation

Our method crucially exploits the annotated sentences in the FrameNet database together with the output of a dependency parser. Our guiding assumption is that sentences that share dependency relations will also share semantic roles as long as they evoke the same or related frames. This is motivated by much research in lexical semantics (e.g., Levin (1993)) hypothesizing that the behavior of words, particularly with respect to the expression and interpretation of their arguments, is to a large extent determined by their meaning. We first describe how predicates are identified and then introduce our model for semantic role labeling.

Predicate Identification Predicate candidates are identified using a simple look-up procedure which compares POS-tagged tokens against FrameNet entries. For efficiency reasons, we make the simplifying assumption that questions have only one predicate which we select heuristically: (1) verbs are preferred to other parts of speech, (2) if there is more than one verb in the question, preference is given to the verb with the highest level of embedding in the dependency tree, (3) if no verbs are present, a noun is chosen. For example, in *Q: Who beat Floyd Patterson to take the title away?*, *beat*, *take away*, and *title* are identified as predicate candidates and *beat* is selected the main predicate of the question. For answer sentences, we require that the predicate is either identical or semantically related to the question predicate (see Section 5).

In the example given above, the predicate *beat* evoques a single frame (i.e., *Cause_harm*). However, predicates often have multiple meanings thus evoquing more than one frame. Knowing which is the appropriate frame for a given predicate impacts the semantic role assignment task; selecting the wrong frame will unavoidably result in erroneous semantic roles. Rather than disambiguiting polysemous predicates prior to semantic role assignment, we perform the assignment for each frame evoqued by the predicate.

Semantic Role Assignment Before describing our approach to semantic role labeling we define dependency relation paths. A relation path R is a relation sequence $\langle r_1, r_2, ..., r_L \rangle$, in which r_l $(l = 1, 2, ..., L)$ is one of predefined dependency relations with suffix of traverse direction. An example of a relation path is $R = \langle subj_U, obj_D \rangle$, where the subscripts U and D indicate upward and downward movement in trees, respectively. Given an unannotated sentence whose roles we wish to label, we assume that words or phrases w with a dependency path connecting them to p are frame elements. Each frame element is represented by an *unlabeled* dependency path R_w which we extract by traversing the dependency tree from w to p. Analogously, we extract from the FrameNet annotations all dependency paths R_{SR} that are *labeled* with semantic role information and correspond to p. We next measure the compatibility of labeled and unlabeled paths as follows:

$$(2) \quad s(w, SR) = \max_{R_{SR} \in M} [sim(R_w, R_{SR}) \cdot P(R_{SR})]$$

where M is the set of dependency relation paths for SR in FrameNet, $sim(R_w, R_{SR})$ the similarity between paths R_w and R_{SR} weighted by the relative frequency of R_{SR} in FrameNet ($P(R_{SR})$). We consider both core and non-core semantic roles instantiated by frames with at least one annotation in FrameNet. Core roles tend to have more annotations in Framenet and consequently are considered more probable.

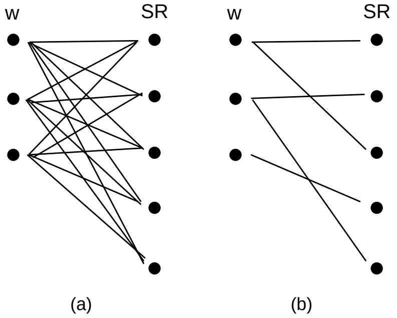

Figure 2: Sample original bipartite graph (a) and its subgraph with edge covers (b). In each graph, the left partition represents frame elements and the right partition semantic roles.

We measure $sim(R_w, R_{SR})$, by adapting a string kernel to our task. Our hypothesis is that the more common substrings two dependency paths have, the more similar they are. The string kernel we used is similar to Leslie (2002) and defined as the sum of weighted common dependency relation subsequences between R_w and R_{SR}. For efficiency, we consider only unigram and bigram subsequences. Subsequences are weighted by a metric akin to $tf \cdot idf$ which measures the degree of association between a candidate SR and the dependency relation r present in the subsequence:

$$(3) \quad weight_{SR}(r) = f_r \cdot \log\left(1 + \frac{N}{n_r}\right)$$

where f_r is the frequency of r occurring in SR; N is the total number of SRs evoked by a given frame; and n_r is the number of SRs containing r.

For each frame element we thus generate a set of semantic role assignments $Set(SRA)$. This initial assignment can be usefully represented as a complete bipartite graph in which each frame element (word or phrase) is connected to the semantic roles licensed by the predicate and vice versa. (see Figure 2a). Edges are weighted and represent how compatible the frame elements and semantic roles are (see equation (2)). Now, for each frame element w

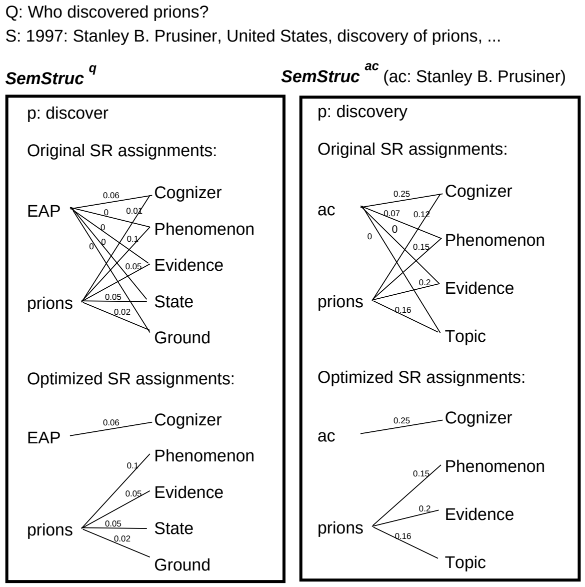

Figure 3: Semantic structures induced by our model for a question and answer sentence

we could simply select the semantic role with the highest score. However, this decision procedure is *local*, i.e., it yields a semantic role assignment for each frame element independently of all other elements. We therefore may end up with the same role being assigned to two frame elements or with frame elements having no role at all. We remedy this shortcoming by treating the semantic role assignment as a *global* optimization problem.

Specifically, we model the interaction between *all* pairwise labeling decisions as a *minimum weight bipartite edge cover problem* (Eiter and Mannila, 1997; Cormen et al., 1990). An edge cover is a subgraph of a bipartite graph so that each node is linked to at least one node of the other partition. This yields a semantic role assignment for all frame elements (see Figure 2b where frame elements and roles are adjacent to an edge). Edge covers have been successfully applied in several natural language processing tasks, including machine translation (Taskar et al., 2005) and annotation projection (Padó and Lapata, 2006).

Formally, optimal edge cover assignments are solutions of following optimization problem:

$$(4) \qquad \max_{E \text{ is edge cover}} \prod_{(nd^w, nd^{SR}) \in E} s(nd^w, nd^{SR})$$

where, $s(nd^w, nd^{SR})$ is the compatibility score between the frame element node nd^w and semantic role node nd^{SR}. Edge covers can be computed efficiently in cubic time using algorithms for the equivalent linear assignment problem. Our experiments used Jonker and Volgenant's (1987) solver.[2]

Figure 3 shows the semantic role assignments generated by our model for the question *Q: Who discovered prions?* and the candidate answer sentence *S: 1997: Stanley B. Prusiner, United States, discovery of prions...* Here we identify two predicates, namely *discover* and *discovery*. The expected answer phrase (EAP) *who* and the answer candidate *Stanley B. Prusiner* are assigned the COGNIZER role. Note that frame elements can bear multiple semantic roles. By inducing a soft labeling we hope to render the matching of questions and answers more robust, thereby addressing to some extent the coverage problems associated with FrameNet.

5 Semantic Structure Matching

We measure the similarity between a question and its candidate answer by matching their predicates and semantic role assignments. Since SRs are frame-specific, we prioritize frame matching to SR matching. Two predicates match if they evoke the same frame or one of its hypernyms (or hyponyms). The latter are expressed by the *Inherits From* and *Is Inherited By* relations in the frame definitions. If the predicates match, we examine whether the assigned semantic roles match. Since we represent SR assignments as graphs with edge covers, we can also formalize SR matching as a graph matching problem.

The similarity between two graphs is measured as the sum of similarities between their subgraphs. We first decompose a graph into subgraphs consisting of one frame element node w and a set of SR nodes connected to it. The similarity between two subgraphs $SubG_1$, and $SubG_2$ is then formalized as:

$$(5) \qquad Sim(SubG_1, SubG_2) = \sum_{\substack{nd_1^{SR} \in SubG_1 \\ nd_2^{SR} \in SubG_2 \\ nd_1^{SR} = nd_2^{SR}}} \frac{1}{|s(nd^w, nd_1^{SR}) - s(nd^w, nd_2^{SR})| + 1}$$

where, nd_1^{SR} and nd_2^{SR} are semantic role nodes connected to a frame element node nd^w in $SubG_1$ and

[2] The software is available from `http://www.magiclogic.com/assignment.html` .

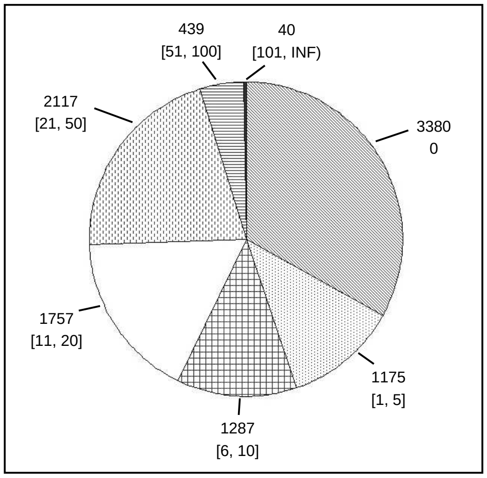

Figure 4: Distribution of Numbers of Predicates and annotated sentences; each sub-pie, lists the number of predicates (above) with their corresponding range of annotated sentences (below)

SubG$_2$, respectively. $s(nd^w, nd_1^{sr})$ and $s(nd^w, nd_2^{SR})$ are edge weights between two nodes in corresponding subgraphs (see (2)). Our intuition here is that the more semantic roles two subgraphs share for a given frame element, the more similar they are and the closer their corresponding edge weights should be. Edge weights are normalized by dividing by the sum of all edges in a subgraph.

6 Experimental Setup

Data All our experiments were performed on the TREC02–05 factoid questions. We excluded NIL questions since TREC doesn't supply an answer for them. We used the FrameNet V1.3 lexical database. It contains 10,195 predicates grouped into 795 semantic frames and 141,238 annotated sentences. Figure 4 shows the number of annotated sentences available for different predicates. As can be seen, there are 3,380 predicates with no annotated sentences and 1,175 predicates with less than 5 annotated sentences. All FrameNet sentences, questions, and answer sentences were parsed using MiniPar (Lin, 1994), a robust dependency parser.

As mentioned in Section 4 we extract dependency relation paths by traversing the dependency tree from the frame element node to the predicate node. We used all dependency relations provided by MiniPar (42 in total). In order to increase coverage, we combine all relation paths for predicates that evoke the same frame and are labeled with the same POS tag. For example, *found* and *establish* are both instances of the frame *Intentionally_create* but the database does not have any annotated sentences for *found.v*. In default of not assigning any role labels for *found.v*, our model employs the relation paths for the semantically related *establish.v*.

Preprocessing Here we summarize the steps of our QA system preceding the assignment of semantic structure and answer extraction. For each question, we recognize its expected answer type (e.g., in *Q: Which record company is Fred Durst with?* we would expect the answer to be an *ORGANIZATION*). Answer types are determined using classification rules similar to Li and Roth (2002). We also reformulate questions into declarative sentences following the strategy proposed in Brill et al. (2002).

The reformulated sentences are submitted as queries to an IR engine for retrieving sentences with relevant answers. Specifically, we use the Lemur Toolkit[3], a state-of-the-art language model-driven search engine. We work only with the 50 top-ranked sentences as this setting performed best in previous experiments of our QA system. We also add to Lemur's output gold standard sentences, which contain and support an answer for each question. Specifically, documents relevant for each question are retrieved from the AQUAINT Corpus[4] according to TREC supplied judgments. Next, sentences which match both the TREC provided answer pattern and at least one question key word are extracted and their suitability is manually judged by humans. The set of relevant sentences thus includes at least one sentence with an appropriate answer as well as sentences that do not contain any answer specific information. This setup is somewhat idealized, however it allows us to evaluate in more detail our answer extraction module (since when an answer is not found, we know it is the fault of our system).

Relevant sentences are annotated with their named entities using Lingpipe[5], a MUC-based named entity recognizer. When we successfully classify a question with an expected answer type

[3] See http://www.lemurproject.org/ for details.

[4] This corpus consists of English newswire texts and is used as the main document collection in official TREC evaluations.

[5] The software is available from www.alias-i.com/lingpipe/

(e.g., *ORGANIZATION* in the example above), we assume that all NPs attested in the set of relevant sentences with the same answer type are candidate answers; in cases where no answer type is found (e.g., as in *Q: What are prions made of?*), all NPs in the relevant answers set are considered candidate answers.

Baseline We compared our answer extraction method to a QA system that exploits solely syntactic information without making use of FrameNet or any other type of role semantic annotations. For each question, the baseline identifies key phrases deemed important for answer identification. These are verbs, noun phrases, and expected answer phrases (EAPs, see Section 3). All dependency relation paths connecting a key phrase and an EAP are compared to those connecting the same key phrases and an answer candidate. The similarity of question and answer paths is computed using a simplified version of the similarity measure[6] proposed in Shen and Klakow (2006).

Our second baseline employs Shalmaneser (Erk and Padó, 2006), a publicly available shallow semantic parser[7], for the role labeling task instead of the graph-based model presented in Section 4. The software is trained on the FrameNet annotated sentences using a standard feature set (see Carreras and Màrquez (2005) for details). We use Shalmaneser to parse questions and answer sentences. The parser makes hard decisions about the presence or absence of a semantic role. Unfortunately, this prevents us from using our method for semantic structure matching (see Section 5) which assumes a soft labeling. We therefore came up with a simple matching strategy suitable for the parser's output. For question and answer sentences matching in their frame assignment, phrases bearing the same semantic role as the EAP are considered answer candidates. The latter are ranked according to word overlap (i.e., identical phrases are ranked higher than phrases with no overlap at all).

7 Results

Our evaluation was motivated by the following questions: (1) How does the incompleteness of FrameNet impact QA performance on the TREC data sets? In particular, we wanted to examine whether there are questions for which in principle no answer can be found due to missing frame entries or missing annotated sentences. (2) Are all questions and their corresponding answers amenable to a FrameNet-style analysis? In other words, we wanted to assess whether questions and answers often evoke the same or related frames (with similar roles). This is a prerequisite for semantic structure matching and ultimately answer extraction. (3) Do the graph-based models introduced in this paper bring any performance gains over state-of-the-art shallow semantic parsers or more conventional syntax-based QA systems? Recall that our graph-based models were designed especially for the QA answer extraction task.

Our results are summarized in Tables 1–3. Table 1 records the number of questions to be answered for the TREC02–05 datasets (Total). We also give information regarding the number of questions which are in principle *unanswerable* with a FrameNet-style semantic role analysis.

Column NoFrame shows the number of questions which don't have an appropriate frame or predicate in the database. For example, there is currently no predicate entry for *sponsor* or *sink* (e.g., *Q: Who is the sponsor of the International Criminal Court?* and *Q: What date did the Lusitania sink?*). Column NoAnnot refers to questions for which no semantic role labeling is possible because annotated sentences for the relevant predicates are missing. For instance, there are no annotations for *win* (e.g., *Q: What division did Floyd Patterson win?*) or for *hit* (e.g., *Q: What was the Beatles' first number one hit?*). This problem is not specific to our method which admittedly relies on FrameNet annotations for performing the semantic role assignment (see Section 4). Shallow semantic parsers trained on FrameNet would also have trouble assigning roles to predicates for which no data is available.

Finally, column NoMatch reports the number of questions which cannot be answered due to frame

[6] Shen and Klakow (2006) use a dynamic time warping algorithm to calculate the degree to which dependency relation paths are correlated. Correlations for individual relations are estimated from training data whereas we assume a binary value (1 for identical relations and 0 otherwise). The modification was necessary to render the baseline system comparable to our answer extraction model which is unsupervised.

[7] The software is available from http://www.coli.uni-saarland.de/projects/salsa/shal/ .

Data	Total	NoFrame		NoAnnot		NoMatch		Rest	
TREC02	444	87	(19.6)	29	(6.5)	176	(39.6)	152	(34.2)
TREC03	380	55	(14.5)	30	(7.9)	183	(48.2)	112	(29.5)
TREC04	203	47	(23.1)	14	(6.9)	67	(33.0)	75	(36.9)
TREC05	352	70	(19.9)	23	(6.5)	145	(41.2)	114	(32.4)

Table 1: Number of questions which cannot be answered using a FrameNet style semantic analysis; numbers in parentheses are percentages of Total (NoFrame: frames or predicates are missing; NoAnnot: annotated sentences are missing, NoMatch: questions and candidate answers evoke different frames.

mismatches. Consider *Q: What does AARP stand for?* whose answer is found in *S: The American Association of Retired Persons (AARP) qualify for discounts...*. The answer and the question evoke different frames; in fact here a semantic role analysis is not relevant for locating the right answer. As can be seen NoMatch cases are by far the most frequent. The number of questions remaining after excluding NoFrame, NoAnnot, and NoMatch are shown under the Rest heading in Table 1.

These results indicate that FrameNet-based semantic role analysis applies to approximately 35% of the TREC data. This means that an extraction module relying solely on FrameNet will have poor performance, since it will be unable to find answers for more than half of the questions beeing asked. We nevertheless examine whether our model brings any performance improvements on this limited dataset which is admittedly favorable towards a FrameNet style analysis. Table 2 shows the results of our answer extraction module (SemMatch) together with two baseline systems. The first baseline uses only dependency relation path information (SynMatch), whereas the second baseline (SemParse) uses Shalmaneser, a state-of-the-art shallow semantic parser for the role labeling task. We consider an answer correct if it is returned with rank 1. As can be seen, SemMatch is significantly better than both SynMatch and SemParse, whereas the latter is significantly worse than SynMatch.

Model	TREC02	TREC03	TREC04	TREC05
SemParse	13.16	8.92	17.33	13.16
SynMatch	35.53^{*}	33.04^{*}	40.00^{*}	36.84^{*}
SemMatch	$53.29^{*\dagger}$	$49.11^{*\dagger}$	$54.67^{*\dagger}$	$59.65^{*\dagger}$

Table 2: System Performance on subset of TREC datasets (see Rest column in Table 1); *: significantly better than SemParse; †: significantly better than SynMatch ($p < 0.01$, using a χ^2 test).

Model	TREC02	TREC03	TREC04	TREC05
SynMatch	32.88^{*}	30.70^{*}	35.95^{*}	34.38^{*}
+SemParse	25.23	23.68	28.57	26.70
+SemMatch	$38.96^{*\dagger}$	$35.53^{*\dagger}$	$42.36^{*\dagger}$	$41.76^{*\dagger}$

Table 3: System Performance on TREC datasets (see Total column in Table 1); *: significantly better than +SemParse; †: significantly better than SynMatch ($p < 0.01$, using a χ^2 test).

Although promising, the results in Table 2 are not very informative, since they show performance gains on partial data. Instead of using our answer extraction model on its own, we next combined it with the syntax-based system mentioned above (SynMatch, see also Section 6 for details). If FrameNet is indeed helpful for QA, we would expect an ensemble system to yield better performance over a purely syntactic answer extraction module. The two systems were combined as follows. Given a question, we first pass it to our FrameNet model; if an answer is found, our job is done; if no answer is returned, the question is passed on to SynMatch. Our results are given in Table 3. +SemMatch and +SemParse are ensemble systems using SynMatch together with the QA specific role labeling method proposed in this paper and Shalmaneser, respectively. We also compare these systems against SynMatch on its own.

We can now attempt to answer our third question concerning our model's performance on the TREC data. Our experiments show that a FrameNet-enhanced answer extraction module significantly outperforms a similar module that uses only syntactic information (compare SynMatch and +SemMatch in Table 3). Another interesting finding is that the shallow semantic parser performs considerably worse in comparison to our graph-based models and the syntax-based system. Inspection of the parser's output highlights two explanations for this. First, the shallow semantic parser has difficulty assigning accurate semantic roles to questions (even when they are reformulated as declarative sentences). And secondly, it tends to favor precision over recall, thus reducing the number of questions for which answers can be found. A similar finding is reported in Sun et al. (2005) for a PropBank trained parser.

8 Conclusion

In this paper we assess the contribution of semantic role labeling to open-domain factoid question answering. We present a graph-based answer extraction model which effectively incorporates FrameNet style role semantic information and show that it achieves promising results. Our experiments show that the proposed model can be effectively combined with a syntax-based system to obtain performance superior to the latter when used on its own. Furthermore, we demonstrate performance gains over a shallow semantic parser trained on the FrameNet annotated corpus. We argue that performance gains are due to the adopted graph-theoretic framework which is robust to coverage and recall problems.

We also provide a detailed analysis of the appropriateness of FrameNet for QA. We show that performance can be compromised due to incomplete coverage (i.e., missing frame or predicate entries as well as annotated sentences) but also because of mismatching question-answer representations. The question and the answer may evoke different frames or the answer simply falls outside the scope of a given frame (i.e., in a non predicate-argument structure). Our study shows that mismatches are relatively frequent and motivates the use of semantically informed methods in conjunction with syntax-based methods.

Important future directions lie in evaluating the contribution of alternative semantic role frameworks (e.g., PropBank) to the answer extraction task and developing models that learn semantic roles directly from unannotated text without the support of FrameNet annotations (Grenager and Manning, 2006). Beyond question answering, we also plan to investigate the potential of our model for shallow semantic parsing since our experience so far has shown that it achieves good recall.

Acknowledgements We are grateful to Sebastian Padó for running Shalmaneser on our data. Thanks to Frank Keller and Amit Dubey for insightful comments and suggestions. The authors acknowledge the support of DFG (Shen; PhD studentship within the International Postgraduate College "Language Technology and Cognitive Systems") and EPSRC (Lapata; grant EP/C538447/1).

References

E. Brill, S. Dumais, M. Banko. 2002. An analysis of the askMSR question-answering system. In *Proceedings of the EMNLP*, 257–264, Philadelphia, PA.

X. Carreras, L. Màrquez, eds. 2005. *Proceedings of the CoNLL shared task: Semantic role labelling*, 2005.

T. Cormen, C. Leiserson, R. Rivest. 1990. *Introduction to Algorithms*. MIT Press.

H. Cui, R. X. Sun, K. Y. Li, M. Y. Kan, T. S. Chua. 2005. Question answering passage retrieval using dependency relations. In *Proceedings of the ACM SIGIR*, 400–407. ACM Press.

T. Eiter, H. Mannila. 1997. Distance measures for point sets and their computation. *Acta Informatica*, 34(2):109–133.

K. Erk, S. Padó. 2006. Shalmaneser - a flexible toolbox for semantic role assignment. In *Proceedings of the LREC*, 527–532, Genoa, Italy.

C. Fellbaum, ed. 1998. *WordNet. An Electronic Lexical Database*. MIT Press, Cambridge/Mass.

C. J. Fillmore, C. R. Johnson, M. R. Petruck. 2003. Background to FrameNet. *International Journal of Lexicography*, 16:235–250.

D. Gildea, D. Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288.

T. Grenager, C. D. Manning. 2006. Unsupervised discovery of a statistical verb lexicon. In *Proceedings of the EMNLP*, 1–8, Sydney, Australia.

R. Jonker, A. Volgenant. 1987. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, 38:325–340.

M. Kaisser. 2006. Web question answering by exploiting wide-coverage lexical resources. In *Proceedings of the 11th ESSLLI Student Session*, 203–213.

J. Leidner, J. Bos, T. Dalmas, J. Curran, S. Clark, C. Bannard, B. Webber, M. Steedman. 2004. The qed open-domain answer retrieval system for TREC 2003. In *Proceedings of the TREC*, 595–599.

C. Leslie, E. Eskin, W. S. Noble. 2002. The spectrum kernel: a string kernel for SVM protein classification. In *Proceedings of the Pacific Biocomputing Symposium*, 564–575.

B. Levin. 1993. *English Verb Classes and Alternations: A Preliminary Investigation*. University of Chicago Press, Chicago.

X. Li, D. Roth. 2002. Learning question classifiers. In *Proceedings of the 19th COLING*, 556–562, Taipei, Taiwan.

D. K. Lin. 1994. PRINCIPAR–an efficient, broad-coverage, principle-based parser. In *Proceedings of the 15th COLING*, 482–488.

D. Moldovan, C. Clark, S. Harabagiu, S. Maiorano. 2003. COGEX: A logic prover for question answering. In *Proceedings of the HLT/NAACL*, 87–93, Edmonton, Canada.

S. Narayanan, S. Harabagiu. 2004. Question answering based on semantic structures. In *Proceedings of the 19th COLING*, 184–191.

S. Padó, M. Lapata. 2006. Optimal constituent alignment with edge covers for semantic projection. In *Proceedings of the COLING/ACL*, 1161–1168.

M. Palmer, D. Gildea, P. Kingsbury. 2005. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

D. Paranjpe, G. Ramakrishnan, S. Srinivasa. 2003. Passage scoring for question answering via bayesian inference on lexical relations. In *Proceedings of the TREC*, 305–210.

S. Pradhan, W. Ward, K. Hacioglu, J. Martin, D. Jurafsky. 2004. Shallow semantic parsing using support vector machines. In *Proceedings of the HLT/NAACL*, 141–144, Boston, MA.

D. Shen, D. Klakow. 2006. Exploring correlation of dependency relation paths for answer extraction. In *Proceedings of the COLING/ACL*, 889–896.

R. X. Sun, J. J. Jiang, Y. F. Tan, H. Cui, T. S. Chua, M. Y. Kan. 2005. Using syntactic and semantic relation analysis in question answering. In *Proceedings of the TREC*.

B. Taskar, S. Lacoste-Julien, D. Klein. 2005. A discriminative matching approach to word alignment. In *Proceedings of the HLT/EMNLP*, 73–80, Vancouver, BC.

M. Wu, M. Y. Duan, S. Shaikh, S. Small, T. Strzalkowski. 2005. University at albany's ilqua in trec 2005. In *Proceedings of the TREC*, 77–83.