The Thirty-First International Florida
Artificial Intelligence Research Society Conference (FLAIRS-31)

SkeletonScore: Guiding a Semantic Parser to Better Results by Example

Ritwik Bose, James Allen
Dept. of Computer Science
University of Rochester
{rbose, james}@cs.rochester.edu

Abstract

The task of semantic parsing is to map natural-language sentences to logical forms representing the underlying meanings of those sentences. Typically, resolving semantic ambiguity is considered to be a side effect of semantic parsing. However a large number of errors in parsing can be attributed to incorrect sense disambiguation in the first place. This can arise from the selection of an incorrect semantic role or semantic type by the parser. This paper applies a knowledge-based algorithm to guide a semantic parser to simultaneously select better semantic types and roles. The algorithm takes into account semantic roles and ontology types to reduce restriction violations arising from incorrect semantic role or type choices, hence increasing the total accuracy of the semantic parser.

Introduction

A first approximation of a semantic parse could be attained by performing the tasks of Semantic Role Labeling (SRL) and Word Sense Disambiguation (WSD). Naturally, these tasks are closely related as the semantic roles a particular word may take are necessarily dependent on their types. A significant portion of errors in semantic parsing can be attributed primarily to incorrect sense assignments. By improving WSD at the constituent level we can guide the semantic parser toward global maxima.

A single incorrect type can result in incorrect roles being assigned to arguments, cascading into further errors. Additionally, arguments to individual predicates often have interdependent sense restrictions. Words with similar meanings naturally share similar argument structures. A particular instance of a sense and its arguments in a semantic parse is a realization of the argument structure of the target sense. We shall call these structures instance predicates. Specifically, instance predicates consist of a single root sense and a set of argument senses associated to the root sense by semantic roles. A single instance predicate acts as a potential template for the argument structure of the target sense and also for the target sense of a particular set of arguments. For any moderately frequent sense there will be many possible instance predicates spanning several argument structures and sense combinations. The tasks of semantic role labeling and word sense disambiguation, while distinct, are interdependent.

In this paper we present a method to use instance predicates extracted from a small number of gold annotated semantic parses to guide the TRIPS parser (Allen, Swift, and de Beaumont 2008; Allen and Teng 2017) to better sense decisions. We adopt an instance based approach to ranking predicates by likelihood based on gold annotated data. Using abstraction along the TRIPS ontology, we are able to elicit an improvement in the TRIPS parser from just a small number of gold predicates.

Motivation

Let us consider statements with two distinct meanings of the word **bass**: `The man played the` **`bass`**`[instrument]` and `The man ate the` **`bass`**`[fish]`. Both are valid sentences, with meanings which are easily disambiguated with just a little bit of common sense knowledge. Rather than attempting to derive the correct meanings of each word through reasoning and a large knowledge base, we would like to capture the essence of whether an interpretation 'sounds plausible' or not. Our abstraction technique allows for statements such as: `A` **`person`** `may use a` **`machine`** and `A` **`person`** `may eat` **`grain`** to correctly identify the required senses by associating bass with machine or grain respectively. Additionally, the same pair of statements can be used to disambiguate bass in the sentence "The bass ate the frog" by partially aligning *bass* with *person*.

Therefore, having an example even in the general vicinity of a correct predicate can produce accurate orderings of interpretations for a statement. The score produced for each interpretation is an estimation of whether the elements in the instance predicate could possibly share the set of relations encoded, based on what has already been seen.

While identifying the precise level and bounds to which a statement can be generalized is a difficult task, collecting instances of such restrictions is not. Every instance of a verb and its arguments represents such a restriction.

The TRIPS Ontology

The TRIPS Ontology is a single inheritance hierarchy organized by a combination of syntactic and semantic attributes.

Each type in the hierarchy contains role restrictions and feature templates. A role restriction for a particular ontology type consist of a semantic role, an abstract ontology type and whether the role described is required or not. These restrictions are hand coded and do not take into account additional preferences induced by combinations of roles. The feature templates indicate various semantic features that an instance of an ontology type may take in a logical form. Since the types are arranged hierarchically, an ontology type may be underspecified, in which case it inherits properties from its parent type.[1]

The TRIPS Parser

The approach presented in this paper relies heavily on the hierarchical structure of the TRIPS ontology. As such, the approach is only directly applicable to the TRIPS parser.

The TRIPS parser is a best-first bottom-up chart parser. The core grammar is a hand-built, lexicalized context-free grammar, augmented with feature structures and feature unification, and driven by a semantic lexicon and ontology. Constituents are produced bottom up and scored using several heuristics to order the constituents by likelihood. At each step, the parser attempts to expand on the most likely constituents on the chart until a single constituent covers the entire input sentence, resulting in a complete parse.[2]

The skeleton-score system refines the ordering of low level constituents in the chart based on the plausibility of the sense structures of each predicate. This in turn leads to an improvement in the final selected parse.

Semantic Skeleton Annotation

The gold annotated logical forms are hand annotated by experts using the procedure described in (Allen et al. 2018). Each sentence is first parsed by the TRIPS parser. The resulting parses are corrected by two experts independently. Finally, the correct parses are reconciled through discussion with a group of experts until a consensus is reached. The process also serves to correct and expand the semantic annotation scheme used by the TRIPS parser as well as identify and correct gaps and errors in the TRIPS ontology.

Related Work

Gildea and Jurafsky (2002) presents a statistical model for semantic role labeling trained on 50,000 hand annotated sentences. However, they find that the task of generalizing the model to unseen predicates remains difficult. Other approaches (Johansson and Nugues 2008; Erk and Pado 2006; Kshirsagar et al. 2015) learn models from FrameNet (Baker, Fillmore, and Lowe 1998) or PropBank (Palmer, Gildea, and Kingsbury 2005). Each resource presents a frame lexicon and sentences annotated to map to those same frames.

Mihalcea and Faruque 2004 describes a minimally supervised sense tagger aimed at using as little annotated data as possible and abstracts word senses up the WordNet (Miller 1995) hypernym hierarchy to compensate for unobserved words.

Banarescu et al. 2013 presented Abstract Meaning Representations (AMR) which are rooted, labeled graphs, compatible with the output of the TRIPS parser. Similarity between AMRs has been computed using the SMatch algorithm (Cai and Knight 2013) which calculates the alignment of nodes between source and target graphs which maximizes the amount of semantic information which is preserved.

Skeleton Score System

The Skeleton Score system takes as an input a predicate and outputs a likelihood score. In this context, a predicate is a subtree of a TRIPS parse with depth 1, consisting of a root type and semantic arguments to that type. Each candidate predicate is compared against a library of gold standard predicates. For this study, the likelihood score is a measure of similarity between the candidate predicate and the most similar gold predicate.

Predicate Similarity

We measure the similarity between two predicates as the average of the element-wise similarity. A valid alignment between two predicates is one that matches each role in the source predicate either to a unique role in the target predicate or a null element.

For two predicates P and Q, $\phi : P \rightarrow Q \cup \{s\}$ is a valid mapping if it satisfies condition $\phi(p_i) = q_j$ only if $role(p_i) = role(q_j)$ and ϕ is one-to-one except at some extra symbol, s. Then,

$$skel(P,Q) = \max_{\phi \in \Phi(P,Q)} \frac{\sum sim_f(type(p_i), type(\phi(p_i)))}{max(|P|,|Q|)} \tag{1}$$

for some similarity measure where $sim_f(x,s) = 0$.

Node Scoring Functions

A node scoring function should return 1 if the two nodes are identical and 0 if they are entirely unrelated. In this study we use three similarity measures: Exact Match, Wu-Palmer similarity and a formulation of cosine similarity. These metrics have desirable property of being distribution agnostic, computing similarity from the structure of the ontology rather than distributions over sense tagged corpora.

Exact Match returns 1 if two nodes are identical and 0 otherwise.

Wu-Palmer Similarity (Wu and Palmer 1994) computes the similarity of two nodes as a ratio of the depth of their least common subsumer and the sum of their depths. For any pair of nodes a given path length apart, the deeper the pair is in the ontology the higher the similarity score is, while node pairs higher up in the ontology with the same path distance receive a lower score. The metric is computed as follows:

$$wup(a,b) = \frac{2 \times d(lcs(a,b))}{d(a) + d(b)} \tag{2}$$

Where $lcs(a,b)$ is the lowest common subsumer of the nodes a and b and d is the depth function.

[1] http://trips.ihmc.us/stripswiki/

[2] http://trips.ihmc.us/parser/cgi/parse

Cosine Similarity is the go-to similarity metric for vector-based NLP applications. However, in the context of a discrete structure, such as an ontology, it is harder to define. In order to compute the cosine of the angle between two nodes in the ontology, we first need to embed the ontology into a vector space. We can embed any tree with k nodes into $\mathbb{R}^{k-1}$ using a function $\rho : V_k \rightarrow \mathbb{R}^{k-1}$ as follows:

$$\rho(v_i)_j = \begin{cases} w_{anc(v_j),v_j} \, if \, v_j \in path(v_0, v_i) \\ 0 \, otherwise \end{cases} \quad (3)$$

That is, the vector representing v_i has in the j^{th} entry the weight of the edge between v_j and its parent if and only if v_j falls on the path between v_i and the root of the tree. It is easy to see how this embedding naturally preserves the $\mathbf{L}_1$ metric. A proof can be found in Bourgain (1985). In this case all edge weights are 1, resulting in $a \cdot b = d(lcs(a, b))$ and $\sum a_i^2 = d(a)$. Thus, we have

$$cosine(a, b) = \frac{d(lcs(a, b))}{\sqrt{d(a)d(b)}} \quad (4)$$

Other considerations

Non-standard values Certain ontology types (e.g. `ELLIPSIS`, `REFERENTIAL-SEM`) do not have direct semantic content. The parser outputs non-ontology types for constructions such as conjunctions and sequences. These types are scored as exact matches. Additionally, the parser may temporarily assign a type as `UNK` to indicate that it will be filled later. These instances receive a score of 0.

Thresholding Since this system only takes positive examples as training examples, it is only possible to find support for candidate predicates. A predicate with a poor score may in fact be a poor candidate or alternatively may have fallen victim to a lack of information. For any node score less than the threshold, we set it to 0 instead. This helps by eliminating mediocre matches and creating a sharper delineation between good and bad matches.

Evaluation

Parser Integration

We integrated the skeleton scoring mechanism into the TRIPS parser in order to perform in-vivo evaluations. On producing a new potential constituent, the parser requests an adjustment factor from the scoring system to modify its internal weight table. The parser continues to pursue the top ranked predicates until a complete parse is found. Hence, the purpose of the skeleton score system is to improve the order in which predicates are expanded. Not every positive adjustment score will result in an actual change to the ordering of the internal weight table, and even when such a change is caused, it may not always result in a change to the final parse.

Given an adjustment range of m the adjustment factor is a value in the range $[1 - m, 1 + m]$ calculated by

$$adj_m(P) = 1 - m + \frac{skel(P)}{2m} \quad (5)$$

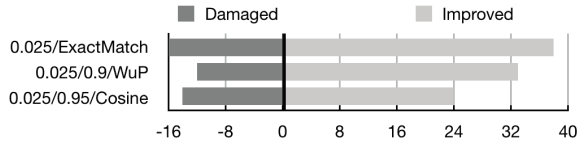

Figure 1: The number of parses either improved or damaged by the Skeleton Score system for the three best performing systems.

Experiments

The dataset consists of a set of 409 sentences and their gold annotated parses. There are a total of 1887 unique gold annotated predicates. Each sentence is parsed with the unmodified "pure" parser for a baseline. The pure parser fails to find a spanning parse for 71 sentences. These sentences are ignored for evaluation purposes. For each sentence, we compare the SMatch score between pure and gold parses to the SMatch score between the skeleton and gold parses.

We use leave-one-out cross-validation. For each sentence we use all predicates occurring in the gold parses of the rest of the corpus as the gold library. For each node scoring function, we vary the adjustment range. We also vary the threshold values for Wu-Palmer and Cosine similarities.

Results

Table 1 shows the relative improvement in SMatch score for each of the experiments. The best improvement is 5.6% over all sentences whose parses are affected by the Skeleton Score system, achieved using Wu-Palmer similarity with the smallest adjustment range of 0.025. In particular, we see relative improvement as the element-wise threshold increases. However, we also see a competing decrease as the adjustment range increases. This is to be expected as the adjustment range dictates the maximum impact the Skeleton Score system can have on the parser. The threshold does not change the maximum magnitude of the impact but does affect the purity of the predicates which create an impact.

	Adjustment Range			
	0.025	0.05	0.1	Threshold
ExactMatch	3.9	2.2	2.1	
WuPalmer	2.9	2.7	1.6	0.85
	5.6	3.9	1.8	0.9
	3.6	3.7	2.9	0.95
Cosine	3.1	2.7	1.6	0.85
	3.5	3.8	2.6	0.9
	3.6	3.6	3.0	0.95

Table 1: The percentage improvement in average SMatch score from the "pure" parser over sentences that are altered by the Skeleton Score system.

We also measure the total impact ratio of each Skeleton Score variant on the final results. We compute this as the ratio of the total number of parses that are altered by the system and the total number of input sentences. This is distinguished over the traditional measure of recall since not every parse needs to be altered by the system and not ever

	Adjustment Range			
	0.025	0.05	0.1	Threshold
ExactMatch	0.257	0.305	0.365	
WuPalmer	0.210	0.188	0.251	0.85
	0.188	0.216	0.285	0.9
	0.162	0.203	0.245	0.95
Cosine	0.225	0.258	0.292	0.85
	0.190	0.213	0.267	0.9
	0.168	0.195	0.238	0.95

Table 2: The impact ratio for each variant of the system

alteration impacts the performance of the system. In table 2 we see a similar pair of tendencies over the impact ratio, where increasing the adjustment range increases impact and increasing threshold decreases it. At the smallest adjustment range, we find that varying the threshold causes more distinct change in the impact ratio than at the highest adjustment range.

Figure 1 shows the number of sentences improved and damaged by each variant of the Skeleton Score system. We note in particular that increasing the size of the adjustment factor increases the total magnitude of the change while increasing the threshold improves the relative quality of the system.

Future Work

The variation in impact ratio and performance suggests that the system does suffer from a lack of annotated data. However, the system is also overall able to avoid negatively impacting the parser by not responding in situations where evidence is too sparse. Hence, future work should first focus on increasing the number of predicates used by the system. The time commitment to annotators and cost of annotation is a barrier which needs to be overcome. To this end, semi-supervised induction of new predicates from resources such as Lore (Gordon and Schubert 2012) may yield results.

Conclusion

We present a knowledge-based algorithm which is able to judge the likelihood of semantic predicates based on annotated examples. With a relatively small library of annotated sentences and predicates we are able to noticeably improve the performance of the TRIPS semantic parser. Our approach integrates positive evidence from an example based approach to the heuristics used in the TRIPS parser to guide parser to better results.

References

Allen, J., and Teng, C. M. 2017. Broad coverage, domain-generic deep semantic parsing. In *AAAI Spring Symposium Series*. AAAI.

Allen, J.; Bahkshandeh, O.; de Beaumont, W.; Galescu, L.; and Teng, C. M. 2018. Effective broad-coverage deep parsing. In *Proc. of the Thirty-Second AAAI Conference on Artificial Intelligence*. New Orleans, Louisiana, USA: AAAI.

Allen, J. F.; Swift, M.; and de Beaumont, W. 2008. Deep semantic analysis of text. In *Proc. of the 2008 Conference on Semantics in Text Processing*, 343–354. Stroudsburg, PA: Association for Computational Linguistics.

Baker, C. F.; Fillmore, C. J.; and Lowe, J. B. 1998. The berkeley framenet project. In *Proc. of the 36th Annual Meeting of the Association for Computational Linguistics*, ACL '98, 86–90. Stroudsburg, PA, USA: Association for Computational Linguistics.

Banarescu, L.; Bonial, C.; Cai, S.; Georgescu, M.; Griffitt, K.; Hermjakob, U.; Knight, K.; Koehn, P.; Palmer, M.; and Schneider, N. 2013. Abstract meaning representation for sembanking. In *Proc. of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, 178–186. Sofia, Bulgaria: Association for Computational Linguistics.

Bourgain, J. 1985. *Some remarks on the Banach space structure of the ball-algebras*. Berlin, Heidelberg: Springer Berlin Heidelberg. 4–10.

Cai, S., and Knight, K. 2013. Smatch: an evaluation metric for semantic feature structures. In *Proc. of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 748–752. Sofia, Bulgaria: Association for Computational Linguistics.

Erk, K., and Pado, S. 2006. Shalmaneser-a flexible toolbox for semantic role assignment. In *Proc. of LREC*, volume 6.

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

Gordon, J., and Schubert, L. K. 2012. Using textual patterns to learn expected event frequencies. In *Proc. of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, 122–127. Stroudsburg, PA, USA: Association for Computational Linguistics.

Johansson, R., and Nugues, P. 2008. Dependency-based semantic role labeling of propbank. In *Proc. of the Conference on Empirical Methods in Natural Language Processing*, 69–78. Association for Computational Linguistics.

Kshirsagar, M.; Thomson, S.; Schneider, N.; Carbonell, J.; Smith, N. A.; and Dyer, C. 2015. Frame-semantic role labeling with heterogeneous annotations. *People* 3:A0.

Mihalcea, R., and Faruque, E. 2004. Senselearner: Minimally supervised word sense disambiguation for all words in open text. In *Proc. of ACL/SIGLEX Senseval-3*.

Miller, G. A. 1995. Wordnet: A lexical database for english. *Commun. ACM* 38(11):39–41.

Palmer, M.; Gildea, D.; and Kingsbury, P. 2005. The proposition bank: An annotated corpus of semantic roles. *Comput. Linguist.* 31(1):71–106.

Wu, Z., and Palmer, M. 1994. Verb semantics and lexical selection. In *Proc. of the 32nd Annual Meeting of the Association for Computational Linguistics*, 133–138. Las Cruces, New Mexico, USA: Association for Computational Linguistics.