A Corpus of Scope-disambiguated English Text

Mehdi Manshadi, James Allen, Mary Swift
Department of Computer Science, University of Rochester
Rochester, NY, 14627, USA
{mehdih,james,swift}@cs.rochester.edu

Abstract

Previous work on quantifier scope annotation focuses on scoping sentences with only two quantified noun phrases (NPs), where the quantifiers are restricted to a predefined list. It also ignores negation, modal/logical operators, and other sentential adverbials. We present a comprehensive scope annotation scheme. We annotate the scope interaction between all scopal terms in the sentence from quantifiers to scopal adverbials, without putting any restriction on the number of scopal terms in a sentence. In addition, all NPs, explicitly quantified or not, with no restriction on the type of quantification, are investigated for possible scope interactions.

1 Introduction

Since the early days of natural language understanding (NLU), quantifier scope disambiguation has been an extremely hard task. Therefore, early NLU systems either devised some mechanism for leaving the semantic representation underspecified (Woods 1978, Hobbs and Shieber 1987), or tried to assign scoping to sentences based on heuristics (VanLehn 1978, Moran 1988, Alshawi 1992). There has been a lot of work since then on developing frameworks for scope-underspecified semantic representations (Alshawi and Crouch 1992, Bos 1996, Copestake et al., 2001, Egg et al., 2001). The motivation of most recent formalisms is to develop a constraint-based framework where you can incrementally add constraints to filter out unwanted scopings. However, almost all of these formalisms are based on hard constraints, which have to be satisfied in every reading of the sentence. It seems that the story is different in practice. Most of the constraints one can hope for (imposed by discourse, pragmatics, word knowledge, etc.) are soft constraints, that is they define a preference over the possible readings of a sentence. As a result, statistical methods seem to be well suited for scope disambiguation.

Surprisingly enough, after two decades of extensive work on statistical techniques in natural language processing, there has not been much work on scope disambiguation (see section 6 for a review). In addition, as discussed later, this work is very restricted. It considers sentences with only two quantifiers, where the quantifiers are picked from a predefined list. For example, it ignores definites, bare singulars/plurals, and proper nouns, as well as negations and other scopal operators.

A major reason for the lack of work on statistical scope disambiguation is the lack of a comprehensive scope-disambiguated corpus. In fact, there is not even a standard test set for evaluation purposes. The reason behind this latter fact is simple. Scope disambiguation is very hard even for humans. In fact, our own early effort to annotate part of the Penn Treebank with full scope information soon proved to be too ambitious.

Instead, we have picked a domain that covers many challenging phenomena in scope disambiguation, while keeping the scope disambiguation fairly intuitive. This helps us to build the first moderately sized corpus of natural language text with full scope information. By fully scoping a sentence, we mean to label the scope interaction between every two scopal elements in that sen-

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics:shortpapers, pages 141–146,
Portland, Oregon, June 19-24, 2011. ©2011 Association for Computational Linguistics

tence. We scope all scope-bearing NPs (quantified or not), negations, logical/modal operators, and other sentential adverbials. We also annotate plurals with their distributive vs. collective readings. In addition, we label sentences with coreference relations because they affect the scope interaction between NPs.

2 Domain

The domain is the description of tasks about editing plain text files; in other words, a natural language interface for text editors such as Linux SED, AWK, or EMACS programs. Figure (1) gives some sentences from the corpus. This domain has several properties that make it a great choice for a first effort to build a comprehensive scope-disambiguated corpus.

First, it carries a lot of scope interactions. As shown in the examples, the domain carries many quantified NPs. Also, scopal operators such as negation, and logical operators occur pretty often in the domain. Second, scope disambiguation is critical for deep understanding in this domain. Third, scoping is fairly intuitive, because a conscious knowledge of scoping is required in order to be able to accomplish the explained task. This is exactly the key property of this domain that makes building a comprehensive scope-disambiguated corpus feasible.

3 Corpus

3.1 The core corpus

The core part of the corpus has been gathered from three different resources, each making up roughly one third of the core corpus.

- *One liners*: These are help documents found on the web for Linux command-line text editors such as SED and AWK, giving a description of a task plus one line of code performing the task.
- *Online tutorials*: Many other online tutorials on using command-line editors and regular expressions exist. Sentences were manually extracted from examples and exercises in these tutorials.
- *Computer science graduate students*: These are the sentences provided by CS graduate students describing some of the routine text editing tasks they often do. The sentences have been provided by both native and non-native English speakers.

1. *Find an occurrence of the word "TBA" in every line and remove it from the line.*
2. *Print a list of the lines that do not start with a digit or end with a letter.*
3. *Replace every string "anti" possibly followed by a hyphen with "not".*

Figure 1. Some examples from the core corpus

3.2 Expanding corpus with crowd sourcing

The core corpus was used to get more sentences using crowd sourcing. We provided input/output (I/O) examples for each task in the core corpus, and asked the workers on Mechanical Turk to provide the description of the task based on the I/O example(s). Figure (2) shows an example of two I/O pairs given to the workers in order to get the description of a single task. The reason for using two I/O pairs (instead of only one) is that there is almost always a trivial description for a single I/O pair. Even with two I/O pairs, we sometimes get the description of a different task, which happens to work for the both pairs. For example the original description for the task given in figure (2) is:

1. *Sort all the lines by their second field.*

The following descriptions are provided by three workers based on the given input/output texts:

2. *Sort the lines alphabetically by the values in the 2nd column.*
3. *Sort the lines by the first group of letters.*
4. *Alphabetize each line using the first letter of each word in the second column.*

(3) gives the description of a different task, but it works for the given I/O pairs. This is not a problem for us, but actually a case that we would prefer to happen, because this way, we not only get a variety of sentences defining the same task, but also obtain descriptions of new tasks. We can add these new tasks to the core corpus, label them with new I/O

INPUT	OUTPUT
1000 NY April 3000 HU August 4000 OR May 4000 AL June	4000 AL June 3000 HU August 1000 NY April 4000 OR May
c josh 21 a adams 23 d sam 26 b john 25	a adams 23 b john 25 c josh 21 d sam 26

Figure 2. Two I/O pairs given for a single task

pairs and hence expand the corpus in a bootstrapping fashion.

The data acquired from Mechanical Turk is often quite noisy, therefore all sentences are reviewed manually and tagged with different categories (e.g. paraphrase of the original description, wrong but coherent description, etc.).

3.3 Pre-processing the corpus

The corpus is tokenized and parsed using the Stanford PCFG parser (Klein and Manning 2003). We guide the parser by giving suggestions on part-of-speech (POS) tags based on the gold standard POS tags provided for some classes of words such as verbs. Shallow NP chunks and negations are automatically extracted from the parse trees and indexed. The resulting NP-chunked sentences are then reviewed manually, first to fix the chunking errors, hence providing gold standard chunks, and second, to add chunks for other scopal operators such as sentential adverbials since the above automated approach will not extract those. Figure (3) shows the examples in figure (1) after chunking. As shown in these examples, NP chunks are indexed by numbers, negation by the letter 'N' followed by a number and all other scopal operators by the letter 'O' followed by a number.

4 Scope annotation

The chunked sentences are given to the annotators for scope annotation. Given a pair of chunks *i* and *j*, three kinds of relation could hold between them.

- *Outscoping constraints:* represented as (*i>j*), which means chunk *i outscopes* (i.e. has a wider scope over) chunk *j*.
- *Coreference relations:* represented as *(i=j)*. This could be between a pronoun and its antecedent or between two nouns.[1]
- *No scope interaction:* If a pair is left unscoped, it means that either there is no scope interaction between the chunks, or switching the order of the chunks results in a logically equivalent formula.

The overall scoping is represented as a list of semicolon-separated constraints. The annotators are allowed to cascade constraints to form a more concise representation (see Figure 3).

1. *Find [1/ an instance] of [2/ the word "TBA"] in [3/ every line] and remove [4/ it] from [5/ the line].*
   (3>1 ; 3=5 ; 1=4) // concise form: (5=3>1=4)
2. *Print [1/ a list] of [2/ the lines] that do [N1/ not] start with [3/ a digit] [O1/ or] end with [4/ a letter].*
   (2>1 ; 2d>N1>3,4 ; N1>O1) // (i>j,k) ≡ (i>j; i>k)
3. *Replace [1/ every string "anti"] [O1/ possibly] followed by [2/ a hyphen] with [3/ "not"].*
   (1>O1>2 ; 1>3)

Figure 3. Chunked sentences labeled with scopings

4.1 Logical equivalence vs. intuitive scoping

Our early experiments showed that a main source of inter-annotator disagreement are pairs of chunks for which, both orderings are logically equivalent (e.g. *two existentials* or *two universals*), but an annotator may label them with outscoping constraints based on his/her intuition. It turns out that the annotators' intuitions are not consistent in these cases. Even a single annotator does not remain consistent throughout the data in such cases. Although it does not make any difference in logic, this shows up as inter-annotator disagreement. In order to prevent this, annotators were asked to recognize these cases and leave them unscoped.

4.2 Plurals

Plurals, in general, introduce a major source of complexity both in formal and computational semantics (Link 1997). From a scope–disambiguation point of view, the main issue with plurals come from the fact that they carry two possible kinds of readings: *collective* vs. *distributive*. We treat plurals as a set of individuals and assume that the index of a plural NP refers to the set (collective reading). However, we also assume that every plural potentially carries an implicit universal quantifier ranging over all elements in the set. We represent this implicit universal with *id* ('*d*' for distributive) where *i* is the index of the plural NP. It is important to notice that while most theoretical papers talk about the collectivity vs. distributivity distinction at the sentence level, for us the right treatment is to make this distinction at the constraint level. That is, a plural may have a collective reading in one constraint but a distributive reading in another, as shown in example 2 in figure (3).

[1] Bridging anaphora relations are simply represented as outscoping relations, because often there is not a clear distinction between the two. However for theoretical purposes, an outscoping constraint (*i*>*j*), where *i* is not *accessible* to *j*, is being understood as a bridging anaphora relation.

4.3 Other challenges of scope annotation

In spite of choosing a specific domain with fairly intuitive quantifier scoping, the scope annotation has been a very challenging job. There are several major sources of difficulty in scope annotation. First, there has not been much work on corpus-based study of quantifier scoping. Most work on quantifier scoping focuses on scoping phenomena, which may be interesting from theoretical perspective, but do not occur very often in practice. Therefore many challenging practical phenomena remain unexplored. During annotation of the corpus, we encountered a lot of these phenomena, which we have tried to generalize and find a reasonable treatment for. Second, other sources of ambiguity are likely to show up as scope disagreement. Finally, very often the disagreement in scoping does not result from the different interpretations of the sentence, but the different representations of the same interpretation. In writing the annotation scheme, extreme care has been taken to prevent these spurious disagreements. Technical details of the annotation scheme are beyond the scope of this paper. We leave those for a longer paper.

5 Statistics

The current corpus contains around 500 sentences in the core level and 2000 sentences acquired from crowd sourcing. The number of scopal terms per sentence is 3.9, out of which 95% are NPs and the rest are scopal operators. Table (1) shows the percentage of different types of NP in the corpus.

The core corpus has already been annotated, out of which a hundred sentences have been annotated by three annotators in order to measure the inter-annotator agreement (IAA). Two of the annotators are native English speakers and the third is a non-native speaker who is fluent in English. All three have some background in linguistics.

5.1 Inter-annotator agreement

Although coreference relations were labeled in the corpus, we do not incorporate them in calculating IAA. This is because, annotating coreference relations is much easier than scope disambiguation, so incorporating them favors toward higher IAAs, which may be deceiving. Furthermore previous work only considers scope relations and hence we do the same in order to have a fair comparison.

Type of NP chunk	*Percentage*
NPs with explicit quantifiers (including indefinite A)	35%
Definites	27%
Bare singulars/plurals	25%
Pronouns	7%
Proper names (files, variables, etc.)	6%

Table 1. Corpus statistics

We represent each scoping using a *directed graph* over the chunk indices. For every outscoping relation $i>j$, node i is connected to node j by the directed edge (i,j). For example, figure (4a) represents the scoping in (5).

5. *Delete [1/ the first character] of [2/ every word] and [3/ the first word] of [4/ every line] in [5/ the file].*
   (5>2>1 ; 5>4>3)

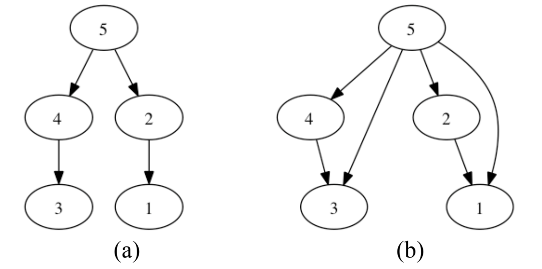
Note that the directed graph must be a DAG (directed acyclic graph), otherwise the scoping is not valid. In order to be able to measure the similarity of two DAGs corresponding to two different scopings of a single sentence, we borrow the notion of *transitive closure* from graph theory. The transitive closure (TC) of a directed graph $G=(V,E)$ is the graph $G^+=(V,E^+)$, where E^+ is defined as follows:

6. $E^+=\{(i,j) \mid i,j \in V \text{ and } i \text{ reaches } j \text{ using a non-null directed path in } G\}$

Given the TC graph of a scoping, every pair (i,j), where i precedes j in the sentence, has one of the following three labels:

- *WS (i outscopes j):* $(i,j) \in E^+$
- *NS (j outscopes i):* $(j,i) \in E^+$
- *NI (no interaction):* $(i,j) \notin E^+ \wedge (j,i) \notin E^+$

A pair is considered a match between two scopings, if it has the same label in both. We define the metrics at two levels, *constraint level* and *sentence level.* At constraint level, every pair of chunks in every sentence is considered one *instance*. At sentence level, every sentence is treated as an in-



(a) (b)

Figure 4. DAG of scoping in (5) and its TC

stance. A sentence counts as a match if and only if every pair of chunks in the sentence has the same label in both scopings. Unlike previous work (section 6) where there is a strong skew in label distribution, in our corpus the labels are almost evenly distributed, each consisting around 33% of the instances. We use *Cohen's kappa score* for multiple annotators (Davies & Fleiss 1982) to measure IAA. Table (2) reports the kappa score.

The IAA defined above serves well for theoretical purposes, but an easier metric could be defined which works fine for most practical purposes. For example, if the target language is first order logic with generalized quantifiers, the relative scope of the chunks labeled NI does not affect the interpretation.[2] Therefore, we define a new version of observed agreement in which we consider a pair a match if it is labeled NI in one scoping or assigned the same label in both scopings. Table (2) reports the IAA based on the latter similarity measure, called κ-EZ.

6 Related work

To the best of our knowledge, there have been three major efforts on building a scope-disambiguated corpus for statistical scope disambiguation, among which Higgins and Sadock (2003) is the most comprehensive. Their corpus consists of 890 sentences from the Wall Street journal section of the Penn Treebank. They pick sentences containing exactly two quantifiers from a predefined list. This list does not include definites, indefinites, or bare singulars/plurals. Every sentence is labeled with one of the three labels corresponding to the first quantifier having wide-scope, the second quantifier having wide scope, or no scope interaction between the two. They achieve an IAA of 52% on this task. The majority of sentences in their corpus (more than 60%) have been labeled with no scope interaction.

Galen and McCartney (2004) is another effort to provide scope-disambiguated data. They pick a set of sentences from LSAT and GRE logic games, which again contain only two quantifiers from a limited list of quantifiers. Their corpus consists of 305 sentences. In around 70% of these sentences, the first quantifier has wide scope. A major problem with this data is that the sentences are artificially constructed for the LSAT and GRE tests.

	Constraint-level	*Sentence-level*
κ	75.0%	66%
κ-EZ	92.3%	89%

Table 2. Inter-annotator agreement

In a recent work Srinivasan and Yates (2009) study the usage of pragmatic knowledge in finding the intended scoping of a sentence. Their labeled data set consists of 46 sentences, extracted from Web1Tgram (from Google, Inc) and hence is open-domain. The corpus consists of short sentences with two specific quantifiers: *Every* and *A*. All sentences share the same syntactic structure, an active voice English sentence of the form *(S (NP (V (NP | PP))))*. In fact, they try to isolate the effect of pragmatic knowledge on scope disambiguation.

7 Summary and future work

We have constructed a comprehensive scope–disambiguated corpus of English text within the domain of editing plain text files. The domain carries many scope interactions. Our work does not put any restriction on the type or the number of scope-bearing elements in the sentence. We achieve the IAA of 75% on this task. Previous work focuses on annotating the relative scope of two NPs per sentence, while ignoring the complex scope-bearing NPs such as definites and indefinites, and achieves the IAA of 52%.

The current corpus contains 2500 sentences, out of which 500 sentences have already been annotated. Our goal is to expand the corpus up to twice in size. 20% of the corpus will be annotated and the rest will be left for the purpose of semi-supervised learning. Since world knowledge plays a major role in scope disambiguation, we believe that leveraging unlabeled domain specific data in order to extract lexical information is a promising approach for scope disambiguation. We hope that availability of this corpus motivates more research on statistical scope disambiguation.

Acknowledgments

This work was supported in part by grants from the National Science Foundation (IIS-1012205) and The Office of Naval Research (N000141110417).

[2] Note that any pair left unscoped is labeled NI. Most of these pairs are those whose both orderings are logically equivalent (section 4.1). Besides, we assume all the scopings are valid that is there is at least one interpretation satisfying them.

References

Alshawi, H. (ed.) (1992) *The core language Engine*. Cambridge, MA, MIT Press.

Alshawi, H. and Crouch, R. (1992) *Monotonic semantic interpretation*. In Proc. 30th ACL, pages 32–39.

Bos, J. (1996) *Predicate logic unplugged*. In Proc. 10th Amsterdam Colloquium, pages 133–143.

Copestake, A., Lascarides, A. and Flickinger, D. (2001) *An Algebra for Semantic Construction in Constraint-Based Grammars*. ACL-01. Toulouse, France.

Davies, M. and Fleiss, J. (1982) *Measuring Agreement for Multinomial Data*. Biometrics, 38:1047–1051,

Egg M., Koller A., and Niehren J. (2001) *The constraint language for lambda structures*. Journal of Logic, Language, and Information, 10:457–485.

Galen, A. and MacCartney, B. (2004). *Statistical resolution of scope ambiguity in Natural language*. http://nlp.stanford.edu/nlkr/scoper.pdf.

Higgins, D. and Sadock, J. (2003). *A machine learning ap-proach to modeling scope preferences*. Computational Linguistics, 29(1).

Hobbs, J. and Shieber, S. M. (1987) *An Algorithm for Generating Quantifier Scopings*. Computational Linguistics 13, pp. 47–63.

Klein, D. and Manning, C. D. (2003). *Accurate Unlexicalized Parsing*. Proceedings of the 41st Meeting of the Association for Computational Linguistics, pp. 423-430.

Link, G. (1998) *Ten Years of Research on Plurals - Where Do We Stand?* Plurality and quantification By Fritz Hamm, Erhard W. Hinrichs, 1998 Kluwer Academic Publishers.

Moran, D. B. (1988). *Quantifier scoping in the SRI core language engine*. In Proceedings of the 26th Annual Meeting of the Association for Computational Linguistics.

Srinivasan, P., and Yates, A. (2009). *Quantifier scope disambiguation using extracted pragmatic knowledge: Preliminary results*. In Proceedings of the Conference on Empirical Methods in Natural Language Processing.

VanLehn, K. (1988) *Determining the scope of English quantifiers*, TR AI-TR-483, AI Lab, MIT.

Woods, W. A. (1978) *Semantics and quantification in natural language question answering*, Advances in. Computers, vol. 17, pp 1-87.