A Probabilistic Treatment of Entity/Event Coercion Ambiguities

Jon Azose

August 13, 2010

Declaration

I hereby confirm that the thesis presented here is my own work, with all assistance acknowledged.

Contents

Chapter 1

Introduction

A ubiquitous problem in computational linguistics is that of ambiguity resolution. In my research I consider a specific type of ambiguity in which words or phrases which represent entities on the semantic level are used to stand in for events. For example, the following two sentences are generally interpreted as being synonymous:

(1) Bill finished the book.

(2) Bill finished *reading* the book.

(as opposed to the second sentence being interpreted to mean throwing the book, sleeping on it, or any of the other infinite range of actions involving the book.) Despite the apparent simplicity of sentence (1), designing a system to recover the missing verb automatically is not a simple endeavor. The problem is further compounded by the fact that the interpretation can change with additional context. For example, the following sentences are both likely to trigger the interpretation of *eating* the book rather than reading it.

(3) My pet goat Bill enjoyed the book.

(4) Bill enjoyed the delicious book made of chocolate.

Successful automated resolution of this type of ambiguity is, for a variety of reasons, a difficult task. Nonetheless, it is a task worth attempting because of the potential benefits. A system with the ability to automatically disambiguate sentences of this nature would be a useful component in tasks such as paraphrase generation and recognizing textual entailment, as well as indirectly providing a wealth of lexical semantic information.

1.1 Research Question

At the most basic level, my research is concerned with a very limited problem. Namely, how to optimally disambiguate sentences which are known to contain this phenomenon in which an entity stands in for an event. However, a number of other considerations naturally came into play in solving this problem. Chief among these considerations is that I wanted to choose a disambiguator design which directly incorporated the lexical semantics inherent in the theoretical approaches to this phenomenon. Further complicating the matter was the initial lack of a real-world training/test set. Therefore, a major component of my work was orchestrating the creation of a gold-standard corpus of disambiguated sentences. (This corpus, of course, is now available for any other researchers wishing to improve on my results.)

1.2 Thesis Outline

The remainder of this document is organized as follows: Chapter 2 contains a review of the literature relevant to this topic. In Chapter 3 I discuss the process of creating the gold standard corpus, including both a detailed refinement of the sentences being considered and the instructions given to human annotators. Chapters 4 through 7 cover the specifications of a baseline disambiguation architecture and three refinements to that architecture: the inclusion of context dependence in Chapter 5, the addition of matching of hand-generated lexical semantic patterns in Chapter 6, and maximum-entropy classification in Chapter 7. Chapter 8 covers two additional minor refinements to the disambiguation architecture. Evaluation of these systems is detailed in Chapter 9. Finally, Chapters 10 and 11 provide conclusions and a discussion of future work.

Chapter 2

Literature Review

2.1 Theory of the Generative Lexicon

The theoretical framework under which my research takes place is Pustejovsky's theory of the generative lexicon [Pustejovsky, 1991], a proposal by Pustejovsky that it is computationally useful to store a variety of semantic and syntactic information together with each entry in a lexicon. To understand how this theory fits in with my research question, it is necessary to first consider more generally the motivations and findings of the theory of the generative lexicon.

A key motivation for this theory is the apparent ability of words to have a variety of semantically related meanings. Consider, for example, the meaning of the word `bake` in the following sentences:

(1) Mary baked the cake.

(2) Mary baked the potato.

In the first sentence, `bake` seems to mean *create via the process of baking* whereas in the latter sentence it means *change via the process of baking.* One way to explain the difference between the two sentences is to posit two different lexical entries for `bake`, $bake_1$ and $bake_2$, corresponding to the two meanings we have seen. Such a solution is undesirable firstly because it ignores the fact that the two meanings of `bake` are closely related, and secondly because it would quickly lead to very large numbers of lexical entries for a single word.

The issue is further complicated by the ability of some verbs to combine with arguments of different syntactic types, as exemplified below:

(3) Peter enjoyed the sandwich.

(4) Peter enjoyed eating the sandwich.

In the former sentence, `enjoy` combines with an NP; in the latter, a VP. If the goal is to be able to provide a compositional means of obtaining the truth conditions of a sentence, this example suggests we may need two different forms of `enjoy`: one which combines with an NP and a second which combines with a VP.

Pustejovsky's contribution is to suggest that rather than multiplying lexical entries for verbs, we keep a minimal number of lexical entries and instead provide a means to convert a component from one syntactic or semantic type to another. He suggests, for example, that `enjoy` should only be allowed to combine with event-denoting arguments. In the case of sentence (3), this requires a means of converting `the sandwich` to the event consisting of eating the sandwich.

Pustejovsky names this conversion process **type coercion**, and defines it as "A semantic operation that converts an argument to the type that is expected by a function, where it would otherwise result in a type error" [Pustejovsky, 1991].

This definition of coercion immediately poses several problems. Firstly, there is the bookkeeping issue of formally defining the coercion function in such a way that logical compositionality is maintained. (As my research does not touch on truth-conditional logical interpretations of sentences, we will simply note this feature of the coercion operation rather than exploring it in detail.) More interestingly, there is the question of where the coercion operation derives its semantics from. For example, in sentence (3), we would like `the sandwich` to be coerced to `eating the sandwich` rather than, say, `throwing the sandwich` or `sitting on the sandwich`.

So how does the coercion operation get access to the correct verb? Pustejovksy's answer is that the necessary information is stored in the lexical entry of the noun. Each word's lexical entry, he suggests, should be equipped with a **qualia structure** consisting of four basic **qualia roles**, each of which contains a different kind of information about the concept represented by the word. The four types of qualia roles he proposes, along with their meanings, are reproduced below: (directly quoted from [Pustejovsky, 1991])

Constitutive Role: the relation between an object and its constituents, or proper parts.

Formal Role: that which distinguishes the object within a larger domain.

Telic Role: purpose and function of the object.

Agentive Role: factors involved in the origin or "bringing about" of an object.

For example, the word `novel` should probably have a telic role of `read` and an agentive role of `write`. In contrast, the word `dictionary` might have telic role `reference` and agentive role `compile`. In a Pustejovskian framework, it is because of this additional semantic information stored in the qualia structure that the coercion operation is able to supply the semantic information which is not explicitly stated in the sentence.

2.2 A Proposed Extension for Exception Handling

[Lascarides and Copestake, 1995] rightly note a number of oversights in Pustejovsky's theory. Most notably, they point out that interpretations which arise as a result of type coercion appear to be overrideable default interpretations. They cite as an example the fact that (5a) means (5b) and not (5c):

(5a) My goat eats anything. He really enjoyed your book.

(5b) The goat enjoyed eating your book.

(5c) The goat enjoyed reading your book.

As a solution to this problem, Lascarides and Copestake propose an extension to Pustejovkian type coercion in which type coercion provides a default interpretation which can then be overridden by pragmatic cues taken from context. They suggest that the ability to override the default coercions be implemented with a combination of discourse representation theory and a wide-domain knowledge base of commonsense entailment relations.

With respect to sentence (5a), they propose that discourse representation theory could tell us that `he` refers to a goat whose propensity for eating is in focus. This, in combination with a fact in the knowledge base that goats are unable to read, would allow the default interpretation (5c) to be overridden by the interpretation in (5b) which is not contradicted by the knowledge base.

2.3 Automated Qualia Structure Acquisition

The ability to intelligently disambiguate sentences in this generative lexical framework requires that we have access to at least partial qualia structures for nouns. As there is no existing large-scale database of words' qualia structures, such structures must be automatically extracted from data. Recent attempts have been made towards automated qualia structure acquisition both from parsed corpus data [Yamada et al., 2007] and from targeted queries on the world wide web [Cimiano and Wenderoth, 2007].

2.3.1 Acquisition from Corpus Data

Given a noun and a list of qualia role candidates, [Yamada et al., 2007] describe two methods for ranking the candidates in order of suitability for the noun's telic and agentive roles. The first method is a maximum entropy learning method. This method involves establishing a small set of known qualia structures and then scanning a parsed corpus on the basis of this training data to learn which syntactic relations are most indicative of a noun's telic and agentive roles. The second method uses hand-generated templates which are thought to be indicative of qualia roles rather than templates obtained by machine learning. These hand generated templates include, for example,

$$\text{N be V[+en]}$$

(as in "This book was written by Chaucer") as a representative template for the agentive role.

A notable aspect of the hand-generated templates Yamada et al. use is the underlying assumption that the target noun occurs as the deep object of a transitive verb. That is, they have no problem determining that the telic role for `book` should be `read` because `book` commonly occurs as the object of `read`. Conversely, this assumption makes them unable to determine that the telic role for `knife` should be `cut` because `knife` commonly occurs as the *subject* for `cut`, not the object.

Furthermore, it should be emphasized that the goal of Yamada et al. is to rank the suitability of a given list of verbs for a noun's qualia structure. They do not address the question of where the verb list should come from in practice; for the purposes of evaluation, they hand-generate a list of 50 verbs for each of 30 nouns. They report results that the one top-ranked role-filler extracted from corpus data was judged to be correct at best 60% of the time.

2.3.2 Acquisition from Web Data

In contrast to the parsed corpus-driven approach of Yamada et al., [Cimiano and Wenderoth, 2007] propose a method for extracting qualia structures on the basis of targeted search engine queries on the world wide web.

Rather than assuming that a list of verbs is given as input, they begin by extracting a list of verbs (or verb phrases) from search engine results. For each qualia role of each noun, several queries are constructed to obtain suitable verb candidates. Query templates include, for example,

to * a(x) new x

in which "x" represents the target noun, "a(x)" represents the indefinite article used with x, and "*" represents the extracted verb or verb phrase. The list of query results are then input to one of a number of information-theoretic ranking methods to determine a ranking.

Cimiano and Wenderoth perform an *a posteriori* evaluation of the quality of their qualia structures. Using the same list of 30 nouns developed by Yamada et al., they asked volunteers to give the quality of the top-ranked verb for a qualia role on a three-point scale. (The scale they use is one in which 0 represents 'wrong', 1 'not totally wrong', 2 'acceptable', and 3 'totally correct' .) They report that their best ranking method was judged to have an average rating of 2.10 on the top-ranked agentive role and 2.16 on the top-ranked telic role [Cimiano and Wenderoth, 2007].

2.4 Probabilistic Treatment of Logical Metonymy

Previous work has been done towards the probabilistic disambiguation of entity/event coercion ambiguities in [Lapata and Lascarides, 2003]. They suggest an elegantly simple method of disambiguation which works only on the basis of counting various verb/object and subject/verb collocations in the British National Corpus. Despite their relatively shallow methods, they achieve good results on an automated paraphrasing task very similar to our automated disambiguation task.

My research addresses two perceived shortcomings with their technique. Firstly, I make an effort to "deepen" the processing by introducing techniques which incorporate patterns targeted towards semantically meaningful telic and agentive relations. Secondly, whereas their evaluation was performed on sentences specifically hand-generated for research in this field, I develop a corpus of such sentences extracted from real-world data and perform my testing on that corpus.

Chapter 3

Data Preparation

3.1 Raw Data and Parsing

The raw data set I use in my research is comprised of a collection of documents taken from Wikipedia and parsed with the Stanford parser.[1] The collection contains a total of approximately 430 million words and represents 80% of the August 2007 Wikipedia dump. [Bouma, 2010]

3.1.1 The Stanford Parser

The Stanford parser is a probabilistic natural language parser implemented in Java. Given an English sentence as input, the Stanford parser provides both a CFG-style parse tree and a typed dependency tree as output. The core architecture was developed by Klein and Manning. More information about the Stanford parser is available in [Klein and Manning, 2003a] and [Klein and Manning, 2003b].

3.2 Identifying Potentially Ambiguous Sentences

I chose to limit my research question to that of being able to provide interpretations for sentences which are known to exhibit an entity/event coercion ambiguity. As I know of no pre-existing collection of such sentences taken from natural language data, it was necessary to first create such a collection by identifying potentially ambiguous sentences within the Wikipedia corpus.

[1] This corpus was provided to me, pre-parsed, by Dr. Gosse Bouma of the University of Groningen.

3.2.1 Selecting a Template for Ambiguity

As mentioned previously, the particular ambiguity I chose to look into was one in which an entity-type noun is coerced into having an event type. Such an ambiguity is known to occur with a variety of verbs, as demonstrated in the examples below, all of which involve a coercion from `book` to the meaning `reading a book`.

(1) I loved your book.

(2) I started your book last night.

(3) That was a very enjoyable book.

Although the first two sentences are syntactically very similar, the third demonstrates that this ambiguity is not restricted purely to sentences in which an entity-NP is in a direct object position.

Thus, it was necessary to isolate a template with which to recognize potentially ambiguous sentences. I chose to restrict myself to sentences in which a verb form of `begin`, `enjoy`, or `finish` is accompanied by a direct object. This choice was made because coercions fitting this template have been particularly well studied in the literature. (See, for example, [Briscoe et al., 1990], [Lascarides and Copestake, 1995], and [Pustejovsky, 1991].)

3.2.2 Extracting Potentially Ambiguous Sentences from the Corpus

Since the Stanford parser gives a typed dependency structure as its output, it is a simple task to isolate those sentences in which `begin`, `enjoy`, or `finish` takes a direct object. The procedure simply consists of scanning the parsed corpus for those sentences containing a dependency of type `dobj` in which the head element is a morphological variant of `begin`, `enjoy`, or `finish`.

However, although sentences extracted in this manner are likely to have the entity/event coercion ambiguity, not all of them do. Prior to using this pattern to extract sentences from the full corpus, I conducted a small-scale feasibility study on 160 extracted sentences which indicated that approximately 20% of corpus sentences matching this pattern contain the desired ambiguity. One common source of undesirable sentences was those in which the direct object already denoted an event. Thus, I put several heuristic measures into place to eliminate the most obvious event-denoting nouns. These measures include the following: disallowing nouns ending in “ion”, which are likely to be event-denoting deverbal nouns (e.g. `construction`, `education`), disallowing nouns ending in “ing”, which are likely to represent misparsed verbs rather than nouns, and removing other common event-denoting nouns not covered by these patterns (e.g. `career`).

In total, this process resulted in the identification of 47,877 candidates for ambiguous sentences

3.3 Human Annotation

Having extracted potentially ambiguous sentences from the corpus, I then asked volunteers to help me annotate those sentences. The volunteers performed two tasks. Firstly, they determined whether a given sentence in fact exhibited the desired ambiguity. Secondly, for those sentences which they deemed to be ambiguous, they provided a verb or verb phrase which, in their opinion, best indicated the correct semantic interpretation for the sentence. These sentences, along with their human-annotated interpretations are taken to be the gold standard for the evaluation of my disambiguation systems. Of the 47,877 candidate sentences, volunteers annotated a total of 6,897 sentences, finding 2,102 ambiguous sentences.

3.3.1 Volunteer Profile

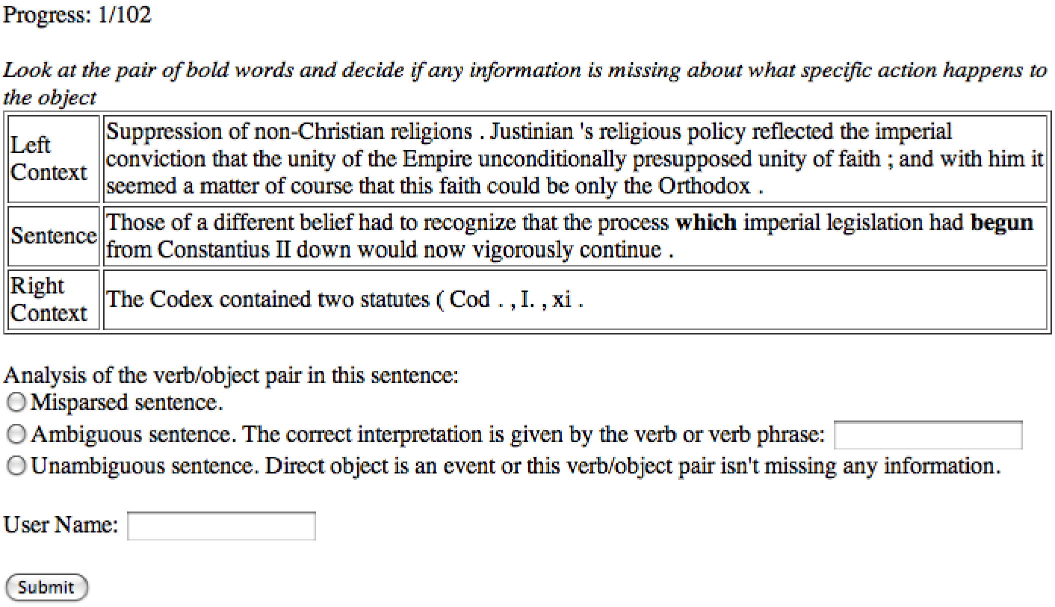
The volunteers who provided annotations were adult, native speakers of American English or adult, proficient non-native English speakers who were studying computational linguistics.

3.3.2 Interface Design

In order to facilitate the annotation task for my volunteers, I implemented an online annotation environment. This environment was coded using HTML, PHP, and JavaScript, and was accessible via the world-wide web

The Annotation Environment

After selecting a file to annotate, volunteers were presented with one ambiguous sentence at a time, as demonstrated in the screenshot in figure 3.1. For added clarity to aid in annotation, each target sentence was accompanied by the two sentences which appear immediately before and after it in the corpus. Volunteers were instructed to select between several annotation options (explained in the following section) and were prompted to supply the “missing” verb for sentences they designated as being ambiguous. Furthermore, the verb-object pair was presented in boldface both to allow identification of misparsed sentences and to facilitate annotation. This same boldface convention is used in the examples in the following section.



Progress: 1/102

Look at the pair of bold words and decide if any information is missing about what specific action happens to the object

Left Context	Suppression of non-Christian religions . Justinian 's religious policy reflected the imperial conviction that the unity of the Empire unconditionally presupposed unity of faith ; and with him it seemed a matter of course that this faith could be only the Orthodox .
Sentence	Those of a different belief had to recognize that the process **which** imperial legislation had **begun** from Constantius II down would now vigorously continue .
Right Context	The Codex contained two statutes (Cod . , I. , xi .

Analysis of the verb/object pair in this sentence:
○ Misparsed sentence.
○ Ambiguous sentence. The correct interpretation is given by the verb or verb phrase:
○ Unambiguous sentence. Direct object is an event or this verb/object pair isn't missing any information.

User Name:

Figure 3.1: Screenshot of the Online Annotation Environment

3.3.3 Instructions to Annotators

Deciding on how best to instruct the volunteer annotators was a non-trivial task. The overarching issue was how to provide a clear, concise set of instructions while still warning the annotators of the most common pitfalls they were likely to encounter. For the purposes of my research, I only truly needed sentences to be divided into two categories: those which contain the desired ambiguity and those which don't. However, in the interest of preparing my annotators to recognize various ways in which a sentence could fail to be ambiguous, I subdivided the "unambiguous sentence" category into four different classes. Reproduced below is the category system into which annotators were requested to place each sentence, as well as several example sentences they were given for each category.

Ambiguous Sentences

Volunteers were instructed to select this option "if the sentence seems to be missing information about precisely what action happens to the object." Examples below include my analysis of the correct interpretation in parentheses after each sentence.

- MTV has confirmed that rapper T-Pain has been asked to produce a few songs for the new album and that Sean Garrett has **finished** three **tracks** . (`recording`)
- He decides to help himself to some , not knowing that Peg-Leg Pete is trying to **enjoy lunch** . (`eating`)
- You may NOT **finish** the **beer** for your teammate . (`drinking`)

Misparsed Sentences

In my opinion, it was neither necessary nor feasible to have volunteers determine whether each target sentence was parsed completely correctly. However, it was important to ascertain that the parser had at least correctly identified a verb/object pair. Volunteers were instructed to select this option if the boldfaced words were not a verb and its direct object. Example sentences receiving this analysis include:

- State-of-the-art Jacquard looms , **finishing equipment** , and cut-and-sew operations were installed .
- New **Beginnings (2006)** .
- Before the summer commences , the two-test series against the Zimbabweans at home and then a triangular one-day tournament in India also involving New Zealand is covered , all **in** which Ponting and his men **enjoyed** great success .

Wrong Meaning of "Enjoy"

This option was presented to volunteers only when annotating sentences with `enjoy`. Quite often in the Wikipedia corpus, `enjoy` means *have the benefit of* rather than *get pleasure from.* Sentences using the former definition do not contain the desired ambiguity, so annotators were requested to note the meaning of `enjoy` in each sentence. Example sentences receiving this analysis include:

- So not only do you **enjoy** the **benefits** of advertising with Clicapic , but you can also make a profit on your ad .
- Al Waseeta famously **enjoyed** only 400 **days** of independence before the assassination of its first and last Amir , Qasim Abdul Rahman .

- Anecdotes of al-Lami 's brutality have spread rapidly throughout Baghdad , and he **enjoys** increasing **support** among Shiite poor .

"Finish" used to report sports results

This option occurs only in sentences with `finish`. In the domain of Wikipedia, `finish` is often used unambiguously and somewhat idiomatically to report sporting results. Example sentences receiving this analysis include:

- In the 2004 Summer Olympics , Komrskov **finished 32nd** in the all-around qualification and was the second alternate for the final as the number of the finalists was reduced from 32 to 24 between 2000 and 2004 Olympic Games .

- Jefferies returned to England for 1988 , playing for Hampshire , who **finished third-bottom** of the 1988 County Championship , the year 's highlight being another of Jefferies ' career total of four ten-wicket match hauls .

- In both the following season the Crusaders **finish runners-up** .

Unambiguous

Finally, I presented volunteers with a catchall option for nouns which denote events or sentences which for any other reason don't contain the desired ambiguity. A literature search turned up no established linguistic tests for determining whether a noun denotes an event. As a general guideline, I instructed my annotators that if the object noun appears to represent an event or is a deverbal noun, or if the correct interpretation seems to be given by a semantically empty verb like "doing", then they should classify the sentence as unambiguous. Example sentences receiving this analysis include:

- In northern Alberta , the Dominion Government **began** a drilling **program** to help define the region s resources .

- Deciding to spend New Year in Azure City , Elan spent quite a while out on the town , **which** he **enjoyed** immensely .

- Bhyrappa completed his primary education in Channarayapatna taluk before moving to Mysore where he **finished** the **rest** of his education .

Chapter 4

Baseline Disambiguation Method

4.1 Technique

Recall that our goal is to disambiguate sentences with a very specific form—namely, those sentences in which some form of `begin`, `enjoy`, or `finish` occurs with a direct object and for which a human annotator has judged the sentence to contain an entity/event coercion ambiguity. Because the theory predicts that the missing semantic information in these sentences is tied to the direct object, a sensible (although not entirely naïve) baseline is to obtain a ranked list of guesses for the "missing verb" by counting verb co-occurrences with the direct object in a training corpus.

More explicitly, the following two-stage process was used to determine a list of guessed interpretations for a given sentence. First we index the corpus and then from that index we extract a list of interpretations.

4.1.1 Corpus Indexing

The goal of this step is to create an index detailing for each noun in a training corpus all the verbs that noun co-occurs with and the absolute frequencies of those co-occurrences. To create this index we look through a Stanford-parsed training corpus for all occurrences of dependencies of type `dobj`. A further restriction is made that the token at the head of the dependency be a verb (i.e. the token must have a Penn Treebank tag beginning with `VB`) and the child of the dependency must be a noun (i.e. have a part of speech tag beginning with `NN`). For each `dobj` relation found, we lemmatize the verb using the MorphAdorner lemmatizer (discussed in section 4.2.1). Then we add this noun-verb pair to the accumulating index.

4.1.2 Suggesting Interpretations

In this step, each ambiguous sentence receives a ranked list of guesses for its “missing verb”. Recall that the ambiguous sentences were all selected on the basis of having `begin`, `enjoy`, or `finish` in a `dobj` relation. To obtain a ranked list for each sentence, we simply take the child from that `dobj` relation and look it up in the index. The corresponding verbs are then ranked on the basis of relative frequency of co-occurrence with that object.

In practice, common nouns generally occur in the corpus with a number of different verbs which is on the order of hundreds. For instance, the noun `novel` occurs in the training corpus as the object of 408 distinct verb types. The most frequent among these verbs are `write` (~27%), `publish` (~12%), `read` (~3%), `complete` (~3%), and `adapt` (~2%).

4.2 Tools

4.2.1 The MorphAdorner Lemmatizer

As mentioned above, the MorphAdorner lemmatizer is used in the baseline disambiguation method to convert verbs to their lemma forms. MorphAdorner is a free, open source tool implemented in Java for performing a variety of morphological annotations on text. These annotations include the ability to adorn a text with standardized spellings, stemmed words, and lemmata [Northwestern University Academic and Research Technologies, 2009]. We make use only of the lemmatizer module which returns lemmatized versions of words.

4.3 Qualitative Error Analysis

After running the baseline disambiguation method, I conducted a qualitative analysis of the errors made by this method on 146 example sentences from pseudo-test data. Specifically, I compared the single top-ranked interpretation output by the baseline disambiguator against the human annotation for the same sentence. I attempted to determine meaningful categories for the types of errors made to highlight different areas with possibilities for improvement over this baseline result. What follows is the classification scheme I devised and the frequency of each type of error.

Full matches (no error) (21.9% of sentences.) These are the sentences in which the top-ranked result from the baseline disambiguation system matches the human-supplied gold-standard result for the same sentence. The verbs need not occur in the same inflection form to be counted as a full match; `reading` and `read`, for example, are considered a match.

Near matches (12.3% of sentences.) Often the gold-standard interpretation is not the only correct way to lexicalize the interpretation of the sentence. If I judged the gold-standard verb and the system-output verb to be semantically equivalent, I classified the sentence as a near match. One example from the data was the sentence "Pan Am ... **began service** to New Zealand on July 12, 1940", which received a human annotation of `offer` whereas the baseline system proposed `provide`.

Annotation is a multi-word expression (11.6% of sentences.) The baseline method is only able to provide one-word interpretations. If the gold-standard annotation was more than one word, this meant an automatic non-match for the baseline system. For example, the sentence "Asimov was a claustrophile; he **enjoyed** small, enclosed **spaces**" received a gold-standard analysis of `be in` whereas the baseline system suggests `have`.

Word sense issues (9.6% of sentences.) This was an issue when the object was the less-common sense of a polysemous noun. For example, "Before that, analog TV had no true 'pixels' to measure horizontal resolution, and vertical scan-line count included off-screen scan lines with no picture information while the CRT beam returned to the top of the screen to **begin** another **field.**" Here the baseline system suggests an interpretation of `have`, likely in response to the more common physical meaning of `field`.

Telic/Agentive transposition (8.2% of sentences.) These are sentences in which the baseline system suggests a response which is a correct telic interpretation for the object when the true interpretation in context should be agentive or vice versa. For example, as mentioned above, the top-ranked verb for `novel` is `write`. However, this will give an incorrect interpretation whenever `novel` appears in a telic context. (e.g. "Bill loves to read. He really **enjoyed** your **novel**.")

Suggested verb is not a telic or agentive verb (7.5% of sentences.) Theoretically, the missing interpretation should be a telic or agentive role of the object. Sometimes the verb suggested by the baseline system doesn't appear to represent either the use of the object or how it was created. In this category were verb guesses like `join` for the object `tour` and `continue` for the object `study`.

Object is not a noun (6.8% of sentences.) Recall that we only indexed cases in which the child of the `dobj` relation was a noun. This excludes cases where the object is a pronoun ("He **enjoyed it**."), a predeterminer ("He **enjoyed all** the books."), or a relative pronoun ("the book **which** he **enjoyed**").

Correct telic/agentive role, but missing contextual cues (6.8% of sentences.) Oftentimes a noun may have different telic or agentive roles depending on who is doing the using or creating. For example, a writer finishes a book by writing it; an editor finishes a book by editing it. These are cases where I judged the top-ranked verb to be *a* correct telic/agentive role for the object, but not *the* correct one for the context.

Top suggestion is begin/enjoy/finish (4.1% of sentences.) The baseline system doesn't exclude the verbs `begin`, `enjoy`, and `finish` from consideration, leading to some redundant annotations when one of those three verbs is the most frequent co-occurrent verb. For example, the top-ranked verb for `pleasures` is `enjoy`, leading to interpretations like "He **enjoyed** enjoying the **pleasures** of life." While this is perhaps a meaningful interpretation of the sentence, it is extremely unlikely to be an interpretation suggested by annotators because of its redundancy.

Named entity issues (3.4% of sentences.) An entertaining example: For the noun `beauty`, we would expect a telic role like `appreciate` and a generic agentive role like `create`. However, the baseline system suggests `sleep`, an idiosyncrasy which is almost certainly due to the relatively high frequency of references to `Sleeping Beauty` in the training corpus. Primarily, issues arose from cases where the Stanford parser failed to distinguish between a common noun and a named entity.

Miscellaneous errors (8.2% of sentences.) This category includes typos, parse errors that slipped through, errors caused by improperly indexing Wikipedia templates, and sentences which volunteers incorrectly annotated as being ambiguous.

Chapter 5

Extension to Integrate Context Dependence

A tempting first addition to the baseline system is to add some weak form of context dependence. The baseline system defined in the previous chapter makes the simplifying assumption that the only information relevant to an ambiguity comes from the object noun. In reality this seems to be a vast oversimplification, so we implement a system which looks at other parts of the sentence as well.

5.1 Motivation

Consider the following three sentences:

(1) The author finished the screenplay.

(2) The reader finished the screenplay.

(3) The editor finished the screenplay.

The interpretations of these three sentences are most likely to be *writing the screenplay*, *reading the screenplay*, and *editing the screenplay* respectively. The fact that this variation is able to occur when the only change to the sentence is in the subject suggests that at the very least it would be helpful to look at the subject of a sentence when disambiguating. In addition to the subject, adjectival modifiers of both the subject and the object may be helpful, at least in theory. Consider:

(4) The man enjoyed the delicious spätzle.

(5) The thirsty man enjoyed the glühwein.

Even if you have never heard of spätzle or glühwein before, you will likely have correctly guessed that spätzle is something you eat and glühwein is something you drink. These interpretations are strongly suggested by the adjectives modifying the object and subject of the sentences.

Given that a sentence's subject and adjectival modifiers may influence its interpretation, the question is then how best to integrate this information into the list of interpretations.

5.2 Technique

The method we use to integrate this additional contextual information is architecturally quite similar to the baseline technique. As before, the process is split into an indexing phase which is only executed once and a suggestion phase which is repeated for each individual sentence.

5.2.1 Corpus Indexing

Previously we had described a system for indexing noun/direct object co-occurrences. Now we simply extend that process to index other types of co-occurrences as well. On the basis of the motivating examples above, we now look at subject/verb co-occurrences as well as verb/adjective pairs. The case of looking at subject/verb co-occurrences should be clear as it is completely analogous with the verb/object indexing technique in the baseline system.

The case of adjectival modifiers is only slightly more complicated. Whereas before we could look within the corpus for all single dependencies of the form

$$verb \rightarrow_{dobj} noun$$

or

$$verb \rightarrow_{nsubj} noun$$

we now must look for two step dependency paths. Specifically, we look for all occurrences in the corpus of

$$verb \rightarrow_{dobj} noun \rightarrow_{amod} adj$$

and

$$verb \rightarrow_{nsubj} noun \rightarrow_{amod} adj$$

We ignore the token-level nouns in these latter patterns and only make an index of the verb/adjective pairs in the two different types of patterns.

As before, indexed words are lemmatized using the MorphAdorner lemmatizer (discussed in section 4.2.1).

5.2.2 Suggesting Interpretations

Now we have described four different types of evidence, outlined in figure 5.1. For convenience, we will label these patterns 1 through 4. Further, to clarify the process by which interpretations get ranked, we will introduce some new notation. A ranked list of verbs for a sentence is obtained by assigning a score to each verb and then simply listing the verbs in decreasing order of score. Thus, for a given sentence the task is really coming up with a list of scores s_v. However, with the inclusion of four distinct types of evidence, it makes sense to talk about subscores calculated from each different pattern. We call these subscores $s_{v,i}$. First we will discuss the technique used to calculate subscores and then the technique for combining them into an overall score.

Pattern	Label
$verb \rightarrow_{dobj} noun$	1
$verb \rightarrow_{nsubj} noun$	2
$verb \rightarrow_{dobj} noun \rightarrow_{amod} adj$	3
$verb \rightarrow_{nsubj} noun \rightarrow_{amod} adj$	4

Figure 5.1: Patterns Representing Different Types of Evidence Indexed

In the case of pattern 1, the subscore calculation is completely equivalent to the ranking technique used in the baseline method. That is, we look in the given sentence for the direct object of `begin`, `enjoy`, or `finish` and then assign scores $s_{v,1}$ as the normalized frequency with which the object co-occurs with verb v in the corpus. This, of course, can be looked up directly from the index.

Assigning subscores for the other patterns proceeds in the same fashion. For example, $s_{v,2}$ is calculated by finding the normalized frequency with which the sentence's subject co-occurs with verb v.

There are, however, some caveats. Because of the way we constructed our example sentences, each ambiguous sentence is guaranteed to have a direct object. However, there is no guarantee that the sentence will have a subject or any adjectives. In the case that the sentence has no subject, $s_{v,2}$ is set at 0 for all verbs (and we do the same with patterns 3 and 4 in the cases where no adjectives modify the subject or object.) It may also be the case that more than one adjective modifies the subject or object. If this is the case, we simply sum the subscores obtained by looking up the different adjectives in the index.

Finally, now that we have calculated subscores on the basis of the four types of evidence, we must combine them in some way. The formula we use to compute the final score is

$$s_v = \sum_{i=1}^{4} \lambda_i s_{v,i}$$

The weights λ_i are chosen on the basis of training data so as to achieve optimal results, as described in the following section.

5.2.3 Finding Optimal Weights

At this point, training data comes into play. For training, we use approximately the first third of the gold-standard corpus—some 700 sentences. Given a numerical metric which describes how well a particular set of weights performs on this training data, the question of finding optimal weights is simply a numerical optimization problem. I chose to use a mean inverse rank metric and a Nelder-Mead optimization technique. The motivation for these choices and brief specifications are given in the following section.

Finally, although I have presented this problem as one with four parameters to optimize (λ_1 through λ_4), there are in fact only three degrees of freedom. Simple calculations will show that for any positive constant c the weights $\{\lambda_1, \lambda_2, \lambda_3, \lambda_4\}$ and $\{c\lambda_1, c\lambda_2, c\lambda_3, c\lambda_4\}$ will give the same rankings. In practice this allowed me to fix $\lambda_1 = 1$ and reduce the dimensionality of the optimization task by 1.

5.3 Tools

5.3.1 Mean Inverse Rank Metric

The metric I chose to optimize over was the mean inverse rank of the gold standard annotations. It is calculated as follows: For each disambiguated sentence, look up the position of the gold standard annotation on the ranked list of suggested interpretations. Take the inverse of this number. (For example, if the gold standard annotation is `read` and `read` appears fourth on the list, the inverse rank is 1/4.) The one exception is the case in which the gold standard annotation does not appear on the list of suggestions. In this case, assign the sentence an inverse rank of 0. Averaging these inverse ranks across all sentences gives the mean inverse rank.

Admittedly, this is not the most intuitive way to assign a number to the performance of the disambiguator. A much simpler metric, for example, might just measure the frequency with which the gold-standard annotation appears as the top-ranked interpretation (or among the top n interpretations.) Such a suggestion does indeed provide a meaningful measure of the system's performance, and will be discussed in the evaluation chapter. However, mean inverse rank is preferable for purposes of optimization because it is much more fine-grained. Small changes to the weights tend to produce at least some change in the calculated mean inverse rank whereas they may not with a simpler metric. That is to say, although the function from weights to mean inverse rank is a discrete step function it is approximately continuous. This is preferable for the optimization technique I chose.

5.3.2 The Nelder-Mead Optimization Technique

The Nelder-Mead algorithm is a technique for finding a local maximum of an objective function. It is a so-called "amoeba" method, which works by calculating the value of the objective function at the vertices of a simplex and then moving one or more vertices of the simplex, eventually contracting onto a local maximum point of the function [Nelder and Mead, 1965]. I selected it because of a number of advantages:

- The algorithm makes no requirements about smoothness or differentiability of the function, which is a nice since we are maximizing a discrete step function.

- At most two function evaluations are required at each step of the algorithm, which is advantageous for us because calculation of our objective function is relatively computationally expensive.

- The algorithm was readily available in a free (for non-commercial use) Java implementation.

The particular implementation used was part of Michael Thomas Flanagan's Java Scientific Library [Flanagan, 2010], a scientific computing package which is offered freely for non-commercial purposes.

5.4 Analysis

[Lapata and Lascarides, 2003] implement a system very similar to this one which takes subject data into account for disambiguation of coercion ambiguities. They report that such a system gives significant improvement over a baseline which only looks at object/verb co-occurrences. Thus, we might expect to find a similar improvement over baseline in our system's performance.

In fact, on development data this refinement offered only an extremely slight improvement over baseline and was subsequently abandoned. It would be wise to ask why they found an improvement while I did not.

Our systems, although based on the same linguistic motivation, use slightly different calculations and it is of course possible that this is one source of the difference in results. However, I believe the majority of the discrepancy is caused by the data being used for evaluation. Their evaluation was performed by checking system output against a gold standard comprised of paraphrase data for hand-generated sentences. These sentences had been generated by [McElree et al., 2001] for a paraphrasing study. Sample sentences include:

(6) The writer finished the novel.

(7) The soldier attempted the mountain.

(8) The teenager finished the novel.

Note that the subjects of these sentences share some common features. For one thing, they are all relatively common nouns. More importantly, they have been selected specifically because they have some bearing on the interpretation of the sentence. This is in stark contrast to the Wikipedia corpus data against which I evaluate. In my corpus it is extremely common for sentence subjects to be either pronouns or named entities. In the case of pronominal subjects, almost nothing can be inferred about the interpretation of the overall sentence by looking at the pronoun. On the other hand, when the subject is a named entity, the number of references to that entity in the corpus used for indexing is likely to be extremely small. This means the evidence coming from the subject pattern is often noisy or insufficient.

In any case, I expect that my technique would achieve similar results to Lapata and Lascarides' system on their test set (although I have not confirmed this.) However, as mentioned before, since this technique failed to produce a significant numerical improvement on my own development set, it was abandoned in favor of other techniques.

Chapter 6

Extension to Target Telic and Agentive Verbs

We saw in the previous chapter that it turned out to be unhelpful on our data set to directly integrate simple evidence from parts of the sentence other than the object noun. However, even if the object noun is the only facet of the sentence we look at, there may be more productive ways to extract information from that object noun than simply counting collocations. Here we introduce a method that makes use of the Pustejovskian theory that the interpretations for these ambiguous sentences should be tied in with the semantics of the object nouns.

6.1 Motivation

If, theoretically, the interpretations of an ambiguous sentence should correspond to telic or agentive roles of the sentence's object, then we could improve our disambiguation results by attempting to automatically determine partial qualia structures for nouns. Ideally we would also try to classify sentences with respect to whether they should take a telic or agentive interpretation. However, as a first approximation, we leave off the classification and simply look for verbs which are either representative of a noun's telic or agentive role. If successful, this should improve results despite the lack of classification because it would de-emphasize those verbs which are neither telic nor agentive.

Pattern	Example sentence
V[+ing] N	I like reading books.
N (worth|deserving|meriting) V[+ing]	Jane Eyre is a book worth reading.
N to V	This is an entertaining book to read.
N used to V	A dagger is a weapon used to stab.
N be used to V	A knife is used to cut.

Figure 6.1: Telic Patterns

Pattern	Example sentence
N be V[+en]	The book was written by Kim.
to V a new N	He started to write a new book.
to V a complete N	He wanted to compile a complete dictionary.
new N have been V[+ed]	A new book has been written.
complete N have been V[+ed]	A complete dictionary has been compiled.

Figure 6.2: Agentive Patterns

6.2 Technique

The architecture of this system is nearly identical to that described in the previous section. As before, there is an indexing stage during which we index all occurrences of certain patterns within the corpus. The evidence discovered from these patterns is still combined using weights which are chosen to be optimal on training data. The only difference is the patterns which are being indexed.

Recall that [Yamada et al., 2007] and [Cimiano and Wenderoth, 2007] both describe techniques for automated acquisition of partial qualia structures. Both teams' techniques revolve around searching for hand-generated patterns which are selected because they are intuitively indicative of nouns' telic and agentive roles. From the list of patterns they describe, I chose 10 and indexed the occurrences of those patterns in the corpus. There were more than 10 patterns presented by these two teams, but some patterns were excluded when preliminary implementations showed them to be extremely infrequent in the Wikipedia corpus. These new patterns are shown in figures 6.1 and 6.2. Taken together with the baseline pattern, this gave a combined total of 11 types of evidence to combine for each sentence and 10 weights to learn. These weights were again learned by numerical optimization with the Nelder-Mead simplex method described in the previous chapter.

6.3 Analysis

Upon finding optimized weights for development data, I discovered that of the ten weights assigned to these patterns, six were negative and two more were very close to 0. The two patterns with significantly positive weights were [V[+ing] N] and [N be V[+en]]; perhaps not coincidentally, these patterns occurred much more frequently in the corpus than the others being considered. This could, of course, indicate that the patterns with low or negative weights are truly counterindicative of correct interpretations and should therefore continue to be considered despite their low or negative weights. However, given the fact that all patterns were selected by hand precisely because they should be meaningful patterns in this context, this explanation seems unlikely. More likely these patterns are given negative weights because of statistical noise or a fluke in the optimization process. (e.g. If the co-occurrence data contained in Pattern A's index is very highly correlated with that of Pattern B's, then a negative weight assigned to Pattern A could simply be compensation for an overly high weight for Pattern B or vice versa.)

Whatever the cause of the negative and near-zero weights, I tested on development data whether those pattern which were initially assigned low weights could be safely removed without an adverse impact on performance. Removing those eight patterns and retraining with only two of the ten patterns produced a change in mean inverse rank of less than 0.001, indicating a certain amount of redundancy of patterns in this context. Thus, the final version of this extension only extends the system with two of the ten patterns—[V[+ing] N] and [N be V[+en]].

Chapter 7

Extension to Incorporate Telic/Agentive Classification

7.1 Motivation

In the previous chapter we equipped our system with the ability to incorporate data from patterns which are representative of nouns' telic and agentive roles. However, there was no strategy in place to distinguish between the two. For example, any time the disambiguator encountered a sentence where `begin`, `enjoy`, or `finish` takes a direct object of `book`, the list of suggested interpretations would be precisely the same. To illustrate why this is a problem, consider the sentences below:

(1) John Doe is a prolific author. In one year he **finished** three **books**.

(2) Jim Smith is a rather slow reader. In one year he **finished** three **books**.

These sentences illustrate that it is plausible that two identical sentences can produce different interpretations depending entirely on their context. In example (1), the context makes reference to an `author`, a noun whose mere presence suggests agentivity. In contrast, the presence of `reader` in the latter example suggests telicity.

On the basis of examples like this, we might hypothesize that performance could be improved by classifying contexts with respect to whether they are indicative of a telic or agentive interpretation. Post-classification, we could then determine interpretations for a sentence using a system like the one in the previous chapter but with a doubled set of weights—one set of weights for sentences classified as being telic and another for agentive sentences.

7.2 Technique

Although the technique is largely analogous to that used in the previous chapters, we lay out all the steps again for the sake of clarity.

7.2.1 Corpus Indexing

The corpus is indexed in precisely the same manner as the previous chapter. One index is created corresponding to the baseline pattern (all verb/direct object co-occurrences in the corpus) and 10 more indices corresponding to the five telic and five agentive patterns from Chapter 6.

7.2.2 Telic/Agentive Classification

In this phase, our goal is to look at a sentence and its immediate context and attempt to determine whether that context is more likely to correspond to a telic or agentive interpretation. In fact, strict classification is neither desired nor necessary. For each sentence in context we obtain a pair of probabilities (e.g. 61% telic, 39% agentive). These probabilities are determined from a maximum-entropy classifier whose parameter values are obtained using TADM, described below.

The classification process is relatively simple. An event, for the purposes of training and classification, is assumed to consist of a sentence containing an ambiguity as well as a two-sentence context window on either side of the target sentence. Each context window is treated as a bucket of words, with tokens being treated as classification features. Half of the training data (approximately 350 sentences) was set aside to train the classifier, while the other half was used to determine patterns' weights. Because the initial volunteer annotations of sentences don't directly contain information about whether the correct interpretation is telic or agentive in nature, this smaller, 350-sentence training set had to be sorted by hand to include this information.

7.2.3 Suggesting Interpretations

Recall from Chapter 5 that the formula we use to calculate a verb's score as the interpretation for a give sentence has the general form

$$s_v = \sum_{i=1}^{k} \lambda_i s_{v,i}$$

where s_v is the overall score assigned to verb v for a given sentence, i ranges over the pattern labels, and $s_{v,i}$ is the score assigned to verb v solely on the basis of evidence from pattern i.

The main difference now is that we essentially have two disambiguators running in parallel: one telic disambiguator and one agentive. We use now a single baseline pattern (to which we assign the index 0), five telic patterns (indexed 1-5), and five agentive patterns (indexed 6-10). For each input sentence, telic and agentive classification probabilities are also calculated; these we will call p_t and p_a. Finally, the formula we use to calculate a verb's overall score is:

$$s_v = s_{v,0} + p_t \sum_{i=1}^{10} \lambda_{i,t} s_{v,i} + p_a \sum_{i=1}^{10} \lambda_{i,a} s_{v,i}$$

The first term in the sum represents the standard baseline disambiguator. The second term represents a specifically telic modification to the disambiguator, weighted by the classifier's probability estimate of the sentence's telicity. Similarly, the final term specifies an agentive modification to the disambiguator. Note in particular that the set of weights has doubled. Each pattern (other than the baseline pattern) now receives separate weights describing how strong a predictor that pattern is for telic versus agentive interpretations.

7.2.4 Finding Optimal Weights

As before, the λ-values are learned via numerical optimization. However, there are now 20 values to learn rather than 10 and we only use half the training data for this process, the first half having already been used to train the classifier.

7.3 Tools

7.3.1 TADM

TADM, the Toolkit for Advanced Discriminative Modeling [Malouf, 2005], is a C++ toolkit for parameter estimation in a variety of discriminative models, including maximum entropy modeling, for which we use it. It was written by Rob Malouf and is available under the Lesser GNU Public License.

Chapter 8

Minor Refinements

In section 4.3 we discussed a variety of errors made by the baseline classifier. A number of these errors went unaddressed by any of the extensions presented in the chapters that followed. To handle the most egregious and easily-surmounted of these errors, I implemented two small refinements to my disambiguators.

8.1 Lightweight Pronoun Resolution

One of the errors we discussed was the failure to handle pronominal objects (or indeed objects of any type other than `NN`, `NNS`, or `NNP`). To ameliorate this shortcoming, I implemented an extremely lightweight form of pronoun resolution. Whenever the object is a pronoun, we simply look backwards through the sentence for the closest preceding word tagged with `NN`, `NNS`, or `NNP`. If it is known whether the object is singular or plural, we also enforce the requirement that the preceding noun agree with the object in number. Once a nominal object is found, all index lookups occur on the basis of that noun rather than the true syntactic object of `begin`, `enjoy`, or `finish`.

Unsurprisingly, this pronoun resolver chooses an incorrect noun a substantial amount of the time. Nevertheless, this is still guaranteed to provide an improvement over baseline. Since only nouns and proper nouns were included in the indexing stage, any sentence with a non-nominal object automatically received an empty list of interpretations from the baseline system.

8.2 Handling “a variety of X”

One relatively common and easy to handle issue revealed in developmental testing was how to handle objects where the object NP has the form “a variety of X" or "a series of X". In these cases it was obvious that the lexical semantics we are interested in comes primarily from the lower-level NP and rather than from `variety` or `series`. From development data I identified a small number of words exhibiting this phenomenon, including `variety`, `series`, `number`, `sequence`, and `first`. Thus, we make the refinement that whenever the sentential object is of the form "a variety/series/... of X" we proceed as if the head noun of X were the direct object of `begin`, `enjoy`, or `finish`.

Chapter 9

Evaluation

Now that we have laid out the various techniques used to suggest interpretations for our ambiguous sentences, we evaluate their performance using several different measures.

9.1 Gold-Standard Evaluation

I set aside a collection of 708 sentences from the gold standard corpus to serve as a test set. Using this collection I compared performance of my various systems under two different measures: the now-familiar mean inverse rank metric and also a metric in which I look for the presence of the gold standard interpretation in the top n suggestions for each sentence.

9.1.1 Evaluation with the Mean Inverse Rank Metric

In Figure 9.1 we present the mean inverse rank performance of four different systems on our test set. The so-called "Realistic Baseline" uses the technique described in Chapter 4. The system called "T/A Extension" extends that baseline with two additional patterns which target telic/agentive verbs, as described in Chapter 6, as well as adding the minor refinements outlined in Chapter 8. The "Classification Extension" is the system described in Chapter 7, again with Chapter 8's refinements.

One further system is also presented in this table to illustrate the performance level of a baseline system with a different approach. This "Naïve Baseline" is the result of considering the disambiguation purely as a classification problem in the simplest possible way. I first compiled a complete list of the gold standard interpretations on the training set. Then, for each sentence, the naïve baseline system suggests precisely the interpretations from

System	MIR
Naïve Baseline	0.094
Realistic Baseline	0.344
T/A Extension	0.393
Classification Extension	0.364

Figure 9.1: Mean Inverse Rank Performance Across Systems

this known list of words, shuffled proportionally to their frequency on the list. (e.g. If 6% of the sentences in the training set had an interpretation of `write`, then `write` would have a 6% chance of occurring first in the list of suggestions.) This is essentially equivalent to treating the disambiguation as a classification problem with no features at all.

As expected, the naïve baseline performs quite poorly compared to the other systems, all of which use co-occurrence data learned from the Wikipedia corpus.

The only real surprise in these results is that the inclusion of a telic/agentive classification component decreases the performance of the system. Several weaknesses caused by the introduction of the classification process stand out as possible reasons for this decrease in performance.

Firstly, by including telic/agentive classification we now must both train a classifier and find optimal pattern weights on the basis of the same training data. In practice, this meant splitting my training data in half, using approximately 350 sentences for each of the two training tasks. (This doubled training could be accomplished instead with a leaving-one-out approach without overtaxing the data so badly, but time constraints restricted me from doing so.)

Secondly, manual inspection of the classifier output suggests that the classifier we use is rather unreliable. Testing on development data in which we artificially substitute a perfect classifier into the process suggests that perfect classification would add an improvement of approximately 0.03 to the mean inverse rank. This demonstrates that classification of contexts into telic or agentive categories is a potentially useful component of the disambiguator if classification can be performed reliably enough. The fact that we get no such boost can be seen as indirect evidence of the weakness of our classifier.

9.1.2 Evaluation with a Top-n Metric

The mean inverse rank metric, although it works well as a fine-grained metric for optimization purposes, is a rather unintuitive measure of a system's

n	Realistic Baseline	T/A Extension	Classification Extension
1	0.253	0.278	0.250
3	0.393	0.465	0.434
10	0.530	0.606	0.583
50	0.695	0.756	0.751

Figure 9.2: Top-n Performance for Various Values of n

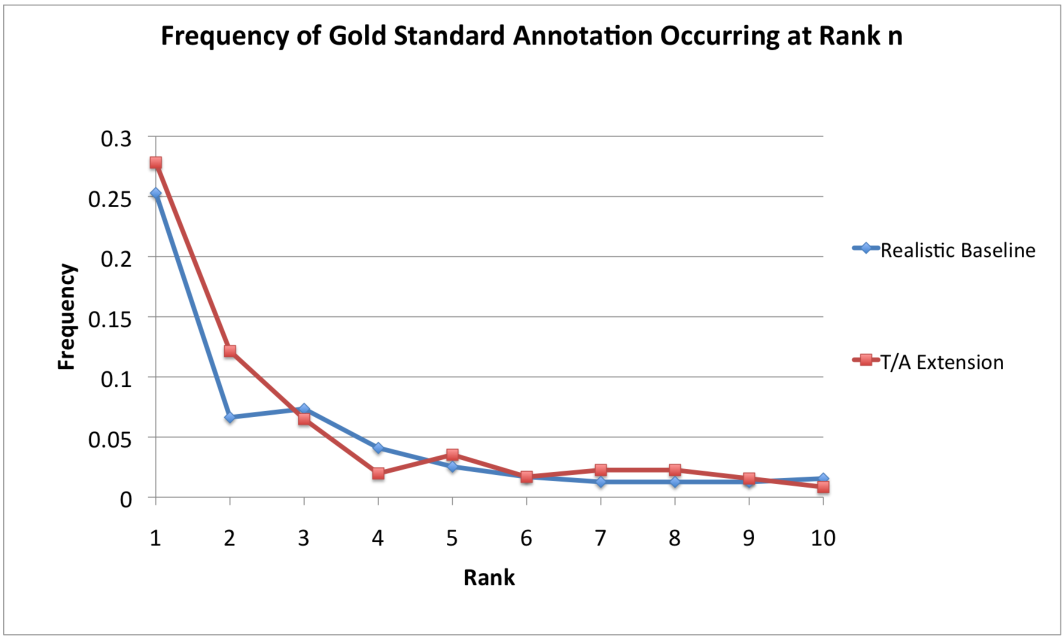


Figure 9.3: Frequency of Gold Standard Evaluations Occurring Precisely at Rank n for the Realistic Baseline and T/A Extension Systems

performance. We might instead ask simply how often the gold standard annotation occurs within the top n suggested interpretations for each system. The table in figure 9.2 answers this question for several values of n.

The good news here is that my best-performing system seems to do quite a good job of concentrating the gold standard interpretations toward the top of its suggestion list. This suggests that my disambiguation system would at the very least be useful in a semi-automated environment, as annotators universally reported that judging whether a given annotation was correct was an easier task than fabricating a correct interpretation from whole cloth.

Looking at the frequency with which the gold standard interpretation occurs precisely at position n gives an even more detailed picture of where exactly we make gains over the baseline system. Such an analysis is graphed in figure 9.3, in which we compare the Realistic Baseline system with the Telic/Agentive Extension. We see, in particular, that the increase in performance over the baseline system is not spread uniformly over all positions in the list, but occurs disproportionately in the top two positions. This is likely to be advantageous for any applications of this system, as bumping up the gold standard annotation from, say, position 12 to position 10 is likely to be much less visible than a bump from position 4 to position 2.

9.2 Qualitative Evaluation

Finally, it is worth noting that the gold standard annotation does not always represent *the* correct lexicalization of an ambiguous sentence's event verb. On the contrary, there are often several lexemes which seem to express correct interpretations for the sentence. Consider, for example, the sentences below:

(1) The scientist began the experiment.

(2a) The scientist began *conducting* the experiment.

(2b) The scientist began *performing* the experiment.

(2c) The scientist began *doing* the experiment.

I personally would judge sentences (2a), (2b), and (2c) to all be acceptable paraphrases of (1). The implication of this is that the evaluations in the previous section in which we only compare results against the gold standard may be underrepresenting the quality of my systems' output, as the gold standard is often just one of several acceptable responses.

In order eliminate the impact of this effect on my evaluation results, I conducted an *a posteriori* evaluation. I gave a collection of 101 sentences from the test set to a single subject. Each sentence was accompanied by the gold standard annotation and the single top-ranked guess from three of my disambiguation architectures. The evaluator was instructed to rate the quality of these four different interpretations on a scale from 0 to 3. The values on the scale were specified as follows: 0 means "wrong", 1 "not totally wrong", 2 "acceptable", and 3 "totally correct". (The scale is the same as that used by [Cimiano and Wenderoth, 2007], who use it for evaluation of qualia structures.) The results of this evaluation are shown in figure 9.4.

We should point out two notable aspects of this graph. Firstly, comparisons against the gold standard data do indeed appear to be underreporting the quality of our results. According to this *a posteriori* evaluation, the

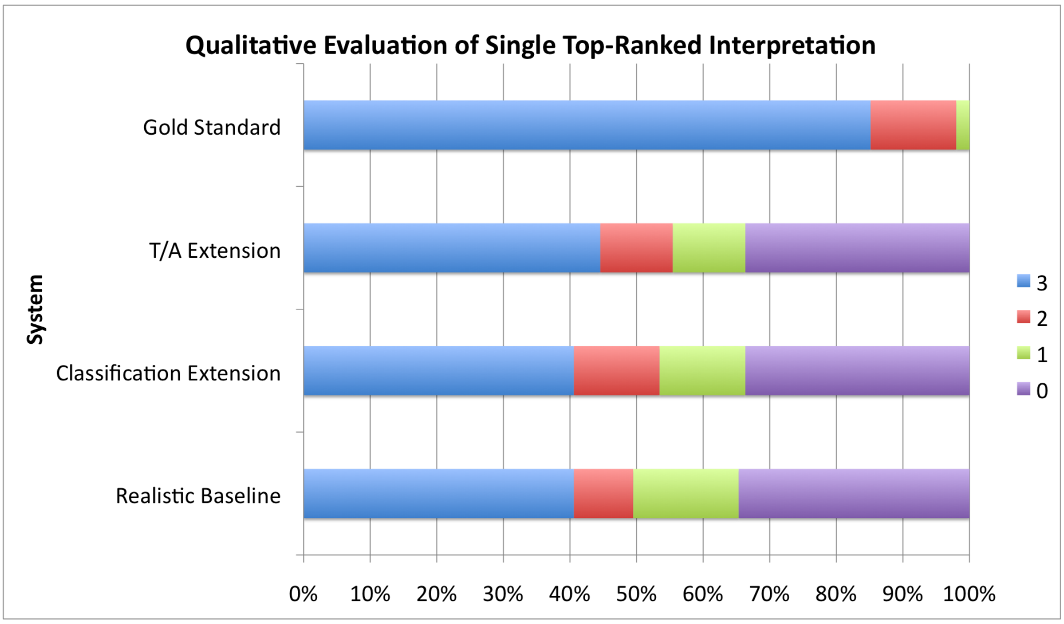


Figure 9.4: Qualitative Evaluation of 101 Sentences on a Three-point Scale

top-ranked verb output by our best performing system is at least acceptable 55.4% of the time. This is considerably better than the 27.8% of the time the same top-ranked verb was identical to the gold standard annotation. Secondly, the gold standard itself appears to be quite reliable. Out of the 101 sentences evaluated, only twice did the gold standard annotation receive a score below 2 (acceptable). This suggests that although volunteer annotators reported the annotation task to be a difficult one, they performed it admirably.

Chapter 10

Conclusions

Motivated by Pustejovsky's generative lexical theory, I set out to perform a targeted disambiguation task—providing the "missing verb" for sentences in which an entity is coerced to an event. In attempting this task, I deliberately chose techniques targeting the lexical semantics of these entity nouns. The techniques included searching a corpus for patterns which were hand-generated to be representative of telic and agentive roles as well as developing a classifier to separate out those sentences with telic interpretations from those with agentive interpretations. Along the way, with cooperation from volunteers, I developed a corpus of disambiguated sentences taken from real-world data. This 2,102-sentence corpus will be made publicly available for use in future research.

The disambiguation system I developed over the course of this research was moderately successful. At best I managed to place the gold standard annotation within the top 10 suggestions for 60.6% of the test set. Perhaps more promisingly, the single top-ranked guess of my best performing system was judged to be "acceptable" or "totally correct" for 55.4% of sentences. These results suggest that I have developed a system with potential to be integrated as a component into tasks like paraphrase generation and recognizing textual entailment.

What is less clear is whether taking an approach driven by lexical semantic considerations brought quantitative improvements over the techniques presented by [Lapata and Lascarides, 2003]. Certainly one advantage of my approach is that it could easily be modified to serve as a partial qualia structure generator, although this application of my techniques will have to be left to future research. Whether the techniques presented herein can be generalized to processing other forms of logical metonymy is left to the readers to consider.

Chapter 11

Future Work

As always seems to be the case with research, I was left with quite a number of statements of the form “If only I had had more time, I would have liked to ...” Here are a some of the most prominent aspects of my research that I consider deserving of future attention.

Firstly, the disambiguator results seem to leave room for improvement. I chose not to focus entirely on techniques which maximize the quantitative results of disambiguation, which means some low-hanging fruit was left unpicked. Examples include, but are not limited to:

Better pronoun resolution: The pronoun resolution component I implemented was extremely crude. For better results, an existing pronoun resolution tool could be used in its place.

Improved telic/agentive classifier: One weak link in the chain was the classifier I put into place to sort contexts with respect to whether the missing verb should be telic or agentive. In fact, I am uncertain whether the telicity or agentivity of a context truly is a kind of discourse phenomenon or merely wishful thinking. Theoretical research into that point would be interesting, as well as having implications to this classification component.

Named entity recognition: Objects which are named entities proved to be a persistent cause of error. To correctly disambiguate a sentence like “He finished The Lord of the Rings”, we would likely need a way of discovering that `The Lord of the Rings` is a novel. This in itself is by no means an easy task.

In short, my quantitative results, being the first results reported on this test set, can almost certainly be improved upon.

Secondly, although I focused primarily on approaches which were driven by the idea of targeting telic and agentive qualia roles, I never got to evaluate my techniques' success in isolating these roles. The final technique I present in Chapter 7 comes temptingly close to offering a fully automated method for telic and agentive role extraction. However, I simply did not have the time to perform the tweaks required to turn my disambiguator into a qualia structure generator. That change could be a first step into new data-driven techniques in automated qualia structure acquisition.

Acknowledgments

Of course a number of people deserve special thanks for their roles in this research. In no particular order:

- Thanks to Gertjan van Noord and Manfred Pinkal for serving as my supervisors.
- To Valia Kordoni, Bobbye Pernice, Gisela Redeker, and Gosse Bouma for all their work coordinating the Erasmus Mundus LCT program, under whose auspices this research took place.
- Thanks to all my volunteer annotators for their excellent work, with special thanks to my most prolific volunteers: Ellen Frierson, Benj Azose, and Arlene Azose.

Bibliography

[Bouma, 2010] Bouma, G. (2010). Personal communication.

[Briscoe et al., 1990] Briscoe, T., Copestake, A., and Boguraev, B. (1990). Enjoy the paper: Lexical semantics via lexicology. In *Proceedings of the 13th International Conference on Computational Linguistics (COLING-90*, pages 42–47.

[Cimiano and Wenderoth, 2007] Cimiano, P. and Wenderoth, J. (2007). Automatic acquisition of ranked qualia structures from the web. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 888–895.

[Flanagan, 2010] Flanagan, M. T. (2010). Java scientific library. Web Address: http://www.ee.ucl.ac.uk/ mflanaga/java/index.html, Last visited: July 9, 2010.

[Klein and Manning, 2003a] Klein, D. and Manning, C. D. (2003a). Accurate unlexicalized parsing. In *IN PROCEEDINGS OF THE 41ST ANNUAL MEETING OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS*, pages 423–430.

[Klein and Manning, 2003b] Klein, D. and Manning, C. D. (2003b). Fast exact inference with a factored model for natural language parsing. In *In Advances in Neural Information Processing Systems 15 (NIPS*, pages 3–10. MIT Press.

[Lapata and Lascarides, 2003] Lapata, M. and Lascarides, A. (2003). A probabilistic account of logical metonymy. *Computational Linguistics*, 29(2):261–315.

[Lascarides and Copestake, 1995] Lascarides, A. and Copestake, A. (1995). The pragmatics of word meaning. In *Journal of Linguistics*, pages 387–414.

[Malouf, 2005] Malouf, R. (2005). Toolkit for advanced discriminative modeling. Web Address: http://tadm.sourceforge.net/, Last visited: July 9, 2010.

[McElree et al., 2001] McElree, B., Traxler, M., Pickering, M., Seely, R., and Jackendoff, R. (2001). Reading time evidence for enriched composition. *Cognition*, 78(1):B17–B25.

[Nelder and Mead, 1965] Nelder, J. and Mead, R. (1965). A simplex method for function minimization. *The computer journal*, 7(4):308.

[Northwestern University Academic and Research Technologies, 2009] Northwestern University Academic and Research Technologies (2009). Morphadorner. Web Address: http://morphadorner.northwestern.edu/morphadorner/, Last visited: May 13, 2010.

[Pustejovsky, 1991] Pustejovsky, J. (1991). The generative lexicon. *Computational Linguistics*, 17.

[Yamada et al., 2007] Yamada, I., Baldwin, T., Sumiyoshi, H., Shibata, M., and Yagi, N. (2007). Automatic acquisition of qualia structure from corpus data. *IEICE - Trans. Inf. Syst.*, E90-D(10):1534–1541.