Mutaphrase: Paraphrasing with FrameNet

Michael Ellsworth and **Adam Janin**
{infinity,janin}@icsi.berkeley.edu
International Computer Science Institute
1947 Center Street, Suite 600
Berkeley, CA 94704-1105 USA

Abstract

We describe a preliminary version of Mutaphrase, a system that generates paraphrases of semantically labeled input sentences using the semantics and syntax encoded in FrameNet, a freely available lexico-semantic database. The algorithm generates a large number of paraphrases with a wide range of syntactic and semantic distances from the input. For example, given the input "I like eating cheese", the system outputs the syntactically distant "Eating cheese is liked by me", the semantically distant "I fear sipping juice", and thousands of other sentences. The wide range of generated paraphrases makes the algorithm ideal for a range of statistical machine learning problems such as machine translation and language modeling as well as other semantics-dependent tasks such as query and language generation.

1 Introduction

A central tenet of statistical natural language processing (NLP) is "there's no data like more data". One method for generating more data is to restate each phrase in a corpus, keeping similar semantics while changing both the words and the word sequence. The efficacy of this approach has been well-established in many areas, including automated evaluation of machine translation systems (Kauchak and Barzilay, 2006), text summarization (Kittredge, 2002), question answering (Rinaldi et al., 2003), document retrieval (Zukerman and Raskutti, 2002), and many others.

Most of the reported work on paraphrase generation from arbitrary input sentences uses machine learning techniques trained on sentences that are known or can be inferred to be paraphrases of each other (Bannard and Callison-Burch, 2005; Barzilay and Lee, 2003; Barzilay and McKeown, 2001; Callison-Burch et al., 2006; Dolan et al., 2004; Ibrahim et al., 2003; Lin and Pantel, 2001; Pang et al., 2003; Quirk et al., 2004; Shinyama et al., 2002). Mutaphrase instead generates paraphrases algorithmically using an input sentence and FrameNet, a freely available lexico-semantic resource (information regarding FrameNet, including relevant terminology, is presented in Section 2).

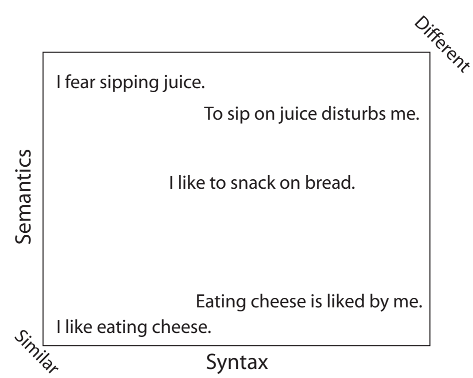

Figure 1: Syntactic and semantic similarity to *I like eating cheese*.

Conceptually, the Mutaphrase algorithm takes a semantic specification of a sentence, provided by an automatic semantic parser such as Shalmaneser (Erk

Proceedings of the Workshop on Textual Entailment and Paraphrasing, pages 143–150,
Prague, June 2007.

and Padó, 2006), and recursively replaces each semantically parsed phrase with a semantically similar phrase. To generate each new phrase, each of the semantic parts of the original phrase is mapped, using FrameNet data, onto a new word or phrase whose position and syntactic marking may be quite different.

The Mutaphrase algorithm outputs a large set of paraphrases with a variety of distances from the input in terms of both syntax and semantics; see Figure 1. Depending on the needs of the application, filtering can be applied to limit the distance to a desired range. For example, language modeling may benefit from a wider variety of semantic outputs, since if *I like eating cheese* is in-domain, then *I like sipping juice* is also likely in-domain. Other applications, e.g. Question Answering, require more stringent limits on semantic distance. See Section 4.

1.1 Current Limitations

The current implementation of Mutaphrase suffers from several limitations. Perhaps the most significant is that the input sentences must be semantically labeled using FrameNet annotations. Since no automated systems for FrameNet-specific annotation are currently incorporated into our algorithm, input is limited to hand-annotated sentences. Also, certain types of semantic ill-formedness are permitted (e.g. *I like sipping meat*), and some types of syntax are not well supported (e.g. conjunctions, relative-clauses). We believe all these factors can be addressed; they are covered briefly in Future Work (Section 4). We confine ourselves in other sections to describing the core Mutaphrase algorithm as currently implemented.

2 FrameNet

The primary resource used in Mutaphrase is FrameNet (Fontenelle, 2003; FrameNet, 2007b), a lexico-semantic database that describes concepts and their interrelations, wordform and word-sequence information, syntactic categories, and mappings between conceptual and lexical/syntactic information. All of these are grounded in hand-annotated examples of real-world sentences. At a slightly more abstract level, FrameNet can be described as providing a two-way mapping between meaning (semantics) and form (syntax, wordforms, sequences).

2.1 Semantics

The conceptual information is represented using **frames**, where a frame is a type of schema or scenario (e.g. Motion, Commercial_transaction), and **frame elements** (FEs), which are the participants and parameters of the frames (e.g. Motion.Path, Commercial_transaction.Buyer). Frames and their frame elements are related and mapped with a limited type of conceptual ontology involving Inheritance (i.e. subtype), Subframe (i.e. temporal subpart), Using (i.e. presupposition) and a few other relation types.

2.2 Syntax

On the form side, the representation is more minimal. **Wordforms** and word-sequences are represented so that words with multiple wordforms (e.g. *take/took*) and word sequences with wordforms (e.g. *take/took off*) can be referred to as unitary objects. We have a category **Support** (and the more specific label 'Copula') for pieces of multi-word expressions that are optional for expressing the semantics of the whole (e.g. *take* in *take a bath*). FrameNet also represents a small but sufficiently rich set of syntactic categories of English (i.e. **phrase types** or PTs, such as 'Sfin', i.e. finite sentence) and syntactic relations (i.e. **grammatical functions** or GFs, e.g. 'Object').

2.3 Syntax-Semantics Bindings

The most vital part of the FrameNet data for our Mutaphrase algorithm is the mappings between semantics and syntax. There are several categories pertaining to this in the data. **Lexical units** (LUs) are a pairing of words/word sequences with the frame each evokes. The **valences** for each LU are sequences in which semantic and form information pertinent to phrases are paired. They are not stored in the database, so we have created a process that produces them entirely automatically (see 3.2). For example, for the LU *hand* in the Giving frame and *possible* in the Likelihood frame, we have the following annotated sentences:

1. [She]$_{Donor/NP/Ext}$ **[handed]**$_{Target}$
   [a bag]$_{Theme/NP/Obj}$
   [to Nob]$_{Recipient/PP(to)/Dep}$

2. [It]$_{Null}$ [was]$_{Copula}$ [**possible**]$_{Target}$ [that he had been hoping to frighten Steve]$_{Hypothetical\_event/Sfin(that)/Dep}$

Example 1 above shows a typical valence, in which most of the positions are semantically labeled with a frame element which is paired with syntactic GF and PT information. The second annotation (2) is more complex, exemplifying each of the major categories that make up the positions of a valence. The categories are:

1. a Null element, with syntax but no semantics (usually *there* or *it*)

2. a Support or Copula with its wordforms

3. a Target (i.e. an LU or word that is part of an LU) with its wordforms, conceptually representing a frame

4. a frame-element/phrase-type/grammatical-function phrase description, which puts together semantic (FE) information with syntax (GF and PT); the PT also indicates fixed words (e.g. the word *that* in the example above)

We can abstract away from the individual sentences, preserving only the sequences of positions with their features, as in the following representation of sentence 2 above:

Null(it), Copula, Target(possible), Hypothetical_event/Dep/Sfin(that)

These abstract valences are the basis for the algorithm we present here. There are typically between two and ten basic patterns associated with each annotated lexical unit, encompassing alternations in the realization of FEs such as Active/Passive (*I recommended her* vs. *She was recommended by me*), the Dative Alternation (*He handed the paper to Stephen* vs. *He handed Stephen the paper*), optional elements (*I ate dinner* vs. *I ate*) and many more.

Basing our algorithm on rearranging the fillers of these FEs allows us to abstract away from syntax, since the FEs of a frame express the same relations regardless of the LU or syntax they occur with. Some meaning differences between LUs within the same frame (e.g. *drink* vs. *eat*) are not overtly modeled in FrameNet. Other resources, such as WordNet, could provide added information in cases requiring finer granularity (see Section 4).

3 Mutaphrase Algorithm

At a very high level, the paraphrase algorithm that we use is as follows: we begin with a sentence with frame-semantic annotation, replace each lexical unit and its associated frame Elements with an alternative valence, then filter the output for its syntactic and semantic fit with the original sentence. The valences may be drawn from either the same LU, an LU of the same frame, or an LU of a related frame.

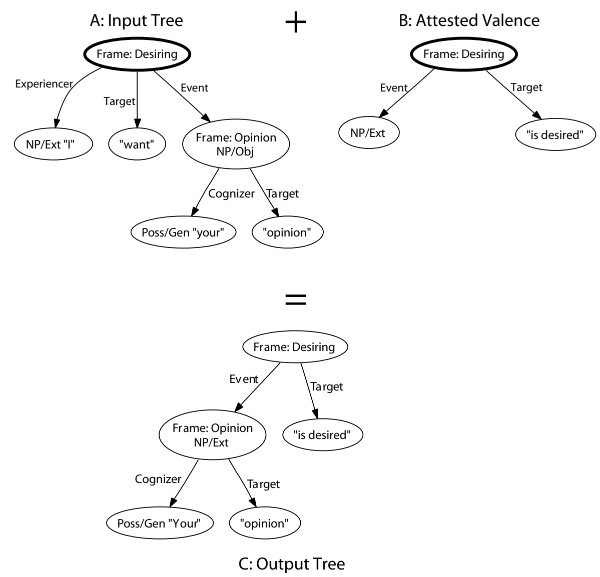

Figure 2: Algorithm Sketch: A syntactic/semantic tree of the original sentence (A) is rearranged to match a different valence (B), producing a new tree (C); thus *I want your opinion* yields the paraphrase *Your opinion is desired*.

Figure 2 shows an example of one step of the algorithm. An input tree for the sentence *I want your opinion* is shown in Figure 2A. The particular valence for the Desiring frame in Figure 2B describes the relations between the word *desire* and its dependents in sentences like *A meeting was desired*. Because the phrase types and grammatical functions of the FEs between the input and the attested valence are compatible, it is possible to replace the input frame with the new valence. The output is shown in Figure 2C.

The remainder of this section describes in more detail how this algorithm is implemented.

3.1 Building a Syntax/Semantics Tree from FrameNet Data

Because the FEs of the original sentence are often filled by phrases with their own annotation, the initial syntactic/semantic annotation is (conceptually, at least) in the form of a graph. Typically, the graph is nearly a tree, with few or no non-tree edges[1]. Hereafter, we will use the term 'tree' even for the cases where there are non-tree edges.

Since the data are not organized in this format in the FrameNet output, we have implemented a routine which can turn FrameNet data into a syntactico-semantic tree; tree examples can be seen in Figure 2A and Figure 2C.

3.2 Building Ordered Valences from FrameNet Data

As mentioned in Section 2.3, we have constructed a routine to parse FrameNet data to produce the valences for each LU of a frame. The basic output is an ordered list of syntactico-semantic elements, optional apositional features (e.g. passive +/-), and the frequency of the pattern.[2]

One innovation of our algorithm is its ability to handle multiword LUs. It simply identifies each word of the LU as a separate element in the list, marking each with the label 'Target'. Thus the ordered valences of *take off.v* in the Undressing frame include, among others:

- Wearer/NP/Ext, take/Target, off/Target, Clothing/NP/Obj; Frequency: 57/68
  (e.g. *I TOOK OFF my watch*)
- Wearer/NP/Ext, take/Target, Clothing/NP/Obj, off/Target; Frequency: 7/68
  (e.g. *You TAKE your shoes OFF*)

One way of thinking about the valence set is that it represents possible orderings of subparts of a phrase that is semantically a frame instance and syntactically a phrase headed by the Target (see, for example, Figure 2B). This semantic/syntactic information is detailed enough to build the syntax of a phrase, given FrameNet-style semantics.

3.3 Core algorithm

Once the input has been turned into a tree and there is a set of alternative ways of expressing each frame that is in the input, the algorithm then recurses downward and then, as it returns up, replaces each phrase/frame node with a set of alternative phrases. In the simplest case, these phrases are built from all the valences that are attested for the frame that the original phrase expressed [3]. In other words, our algorithm is a recursive tree-rewrite in which the current valence of the current LU is replaced by many alternate valences of many different LUs.

In the recursion, word and phrase nodes not headed by an LU are kept the same (except for pronouns, which are expanded to all their wordforms, e.g. *me* to *I/me/my/mine*). The child phrases of such an unparaphrased node, if they are headed by an LU or pronoun, can be paraphrased as long as the paraphrases match the phrase type and grammatical function of the original child phrase.

In Figure 2, the original sentence (represented in Figure 2A) has the phrase representing the Desiring frame replaced with an alternative phrase evoking the same frame (Figure 2B) to produce a new, roughly semantically equivalent sentence (Figure 2C) by expressing the same set of frames in the same FE relations to each other.

In practice, we have to throw away at the outset many of the valences because they include FEs that are not in the input sentence[4] or because they have syntactic requirements of their child phrases which

[1]These non-tree edges are introduced when a phrase is an FE of more than one frame. In keeping with normal syntactic analysis, we treat the node as non-local to all but one parent.

[2]Although frequency of a particular pattern in the FrameNet data is not strictly representative of the frequency of that pattern in the corpus, a close examination reveals that the rank order of patterns is largely identical, i.e. the most common pattern in FrameNet represents the most common pattern in the corpus. How useful this inexact statistical data will be is the subject of future research.

[3]Our algorithm will work just as well with related frames as long as the relevant FEs are mapped in the FrameNet data. Controlling the distance, direction, and relation-types of related frames that are included for paraphrase (if any) is one way to control the degree of semantic diversity of the paraphrase output. See further Section 3.4.

[4]Thus attempting to use the valence Experiencer/NP/Ext, Degree/AVP/Dep, want/Target, Event/NP/Obj (e.g. *I really*

cannot be filled by a paraphrase of the child phrases. For example, for the input sentence *I gave presents to friends*, the code can output 560 (unfiltered) paraphrases. A random selection from the output includes *Presents bequeathed to friends*, *I handed in presents*, and *Presents donated by I*. Of these, the first and last are filtered out as not filling the original sentential context and the last, in addition, is filtered out because of the mismatch between the pronoun wordform *I* and the non-subject grammatical function.

To further refine the paraphrases, we must eliminate examples that are not compatible with the input sentence. In our current implementation, our algorithm filters out incorrect syntax during the recursion over the tree. Ultimately, we will also filter out malformed semantics. The rest of this section is devoted to an explication of the details of this filtering.

3.4 Syntactic/Semantic Compatibility

For both syntax and semantics, the degree of viability of a paraphrase can be divided up into two components: well-formedness and similarity. Syntactic and semantic well-formedness is always desirable and the algorithm seeks to maximize it in ways that are outlined below. Similarity between the original sentence and its paraphrases (or among the paraphrases), however, may be more or less desirable depending on the task. Figure 1 shows an example of the various degrees of syntactic and semantic similarity of the paraphrase output. To maintain flexibility, we will need several control parameters to allow us to filter our output for syntactic/semantic similarity.

3.4.1 Syntactic Compatibility

Syntactic incompatibilities most commonly result from gross mismatches between the Phrase Type called for in a new valence and the Phrase Type possibilities available for the child phrase.

For example, if the initial sentence for paraphrase is *I want your opinion* as in 1 below (repeated from Figure 2), Valence 2 below represents a PT mismatch, since *I*, an NP filler of the Experiencer role in the original sentence, is not modifiable into an adjective phrase (AJP).

1. Experiencer/NP/Ext, want/Target, Event/NP/Obj

2. There/Null, be/Copula, Experiencer/AJP/Dep, desire/Target, Event/PP(for)/Dep
(e.g. *There is a public desire for transparency*)

3. There/Null, be/Copula, desire/Target, Experiencer/PP(in)/Dep, Event/PP(for)/Dep
(e.g. *There was a desire in America for home rule*)

This filtering is vital, as otherwise valence 2 would yield the awful *There is me desire for your opinion*.

However, phrase types that are not exact matches may nevertheless be compatible with each other. Valence 3, for example, is compatible with the original valence, since the original Experiencer and Event FEs were filled by NPs, to which prepositions can be added to match the PP realizations required by Valence 3. This yields another paraphrase of the sentence in Figure 2: *There is a desire in me for your opinion*. Similarly, full sentential clauses can be modified to match VPs by truncation of the External (subject) argument, etc. A phrase from the original sentence may also be omitted to match an empty phrase in the paraphrase, as seen in the omission of the Experiencer in the paraphrase in Figure 2.

These alternations provide more variety in the potential phrase types of the paraphrases. Which syntactic modifications are allowed should be an externally controllable parameter, but this has not yet been implemented. In general, allowing fewer types of modification should move the average output leftward in the syntax/semantic similarity graph in Figure 1 (toward more syntactic similarity).

Although every annotated valence represents a grammatical structure, some of these structures will more likely be judged as well-formed than others; in particular, infrequent patterns are more likely ill-formed than frequent ones. An additional controllable parameter, allowing a trade-off between recall and precision, is a frequency cut-off for accepting a valence pattern based on the number of times

want another chance) when paraphrasing the initial sentence in Figure 2 will not work, since there is nothing in the original to fill the Degree FE mentioned here.

the pattern is found in the FrameNet data. Our algorithm currently produces a ranked list of paraphrases based on exactly this frequency parameter, and downstream processing can choose a cut-off frequency or n-best to reduce the total output.

3.4.2 Semantic Filtering

Lexical units of the same frame are not necessarily synonyms; they may be antonyms or coordinate terms (i.e. co-hyponyms). For example, *cheese* and *juice* are both in the Food frame, but *I like eating cheese* and *I like eating juice* are certainly not a semantic match! In fact, the second is a semantically ill-formed modification of the first. Similarly, *like* and *hate* are both in the Experiencer_subject frame. While *I hate eating cheese* is similar to *I like eating cheese* in describing an attitude toward eating cheese, they are not an exact semantic match either; in this case, however, the lack of semantic similarity does not lead to semantic ill-formedness.

For some tasks such as expanding a language model, exact semantic match is not necessary, but for tasks that require strict semantic match, there are several simple ways to increase robustness.

Tighter filtering, of whatever kind, will move the average output of the algorithm downward in the syntax/semantic similarity graph in Figure 1 (toward more semantic similarity).

3.5 Preliminary Results

We have implemented the above algorithm to the point that it is capable of producing paraphrases of arbitrary input sentences that have received proper FrameNet annotation. A large number of paraphrases with a variety of phrase types are produced, but the lack of semantic filtering occasionally leads to semantically ill-formed results. The output is ranked purely according to the frequency in the FrameNet data of the valences used to build the paraphrase.

For the sentence *I like eating cheese*, the paraphraser produced 8403 paraphrases, of which the following was top-ranked: *I resented drinking cheese*, which suffers from the semantic mismatch problems discussed in Section 3.4.2. Some other output at random:

- I am interested in cheese devouring.
- I was nervous that cheese's ingested.
- I'm worried about gobbling down cheese.
- My regrets were that cheese was eaten by me.

Since most of the annotation in the Ingestion frame (the frame for *eat*, etc.) concerns eating rather than drinking, the majority of the output is semantically well-formed. The paraphrases generated from the Experiencer_subject frame (the frame for *like, interested, regret*, etc.) are more uniformly felicitous, even if semantically quite divergent from the meaning of the original. Both the infelicity of *drinking cheese* and the semantic divergence appear to be addressable by refining semantic tightness using WordNet. Averaging over senses, words like *gobble* and *ingest* have lower WordNet-based semantic distance from *eat* than *drink*.

For the sentence *Nausea seems a commonplace symptom*, the paraphraser outputs 502 paraphrases, of which the following was top-ranked: *It seems a commonplace sign*. Other output at random:

- Tiredness looks indicative.
- Queasiness smelt of a commonplace sign.
- Sleepiness appears a commonplace sign.
- Queasiness smelt indicative queasiness.
- Somnolence appears to be indicative.

Longer sentences (e.g. *Locally elected school boards, especially in our larger cities, become the prey of ambitious, generally corrupt, and invariably demagogic local politicians or would-be politicians*) currently take excessive amounts of time and memory to run, but typically produce 10,000+ paraphrases. Pruning earlier during paraphrase generation should help address this issue.

4 Future Work

Currently, Mutaphrase requires the input sentences to have been marked with FrameNet annotations prior to processing. Although automatic semantic parsing is a large and growing field (Moldovan et al., 2004; Litkowski, 2004; Baldewein et al., 2004), two problems present themselves. First, output from an automated parser is not typically compatible with FrameNet markup. Although this is mostly "a simple matter of programming", some linguistic tools must be developed to convert between formats (e.g. to infer FrameNet phrase types from part-of-speech tags).[5] Second, it is not yet clear how the inevitable errors introduced by the parser will affect the Mutaphrase algorithm[6]. We plan to use application-dependent measures to judge the effects of parsing errors.

Certain types of semantic ill-formedness cannot be detected by the current version of Mutaphrase. A typical example is *I like sipping beef* as a paraphrase of *I like eating cheese*. We can guarantee semantic well-formedness by limiting paraphrases to morphologically related words (e.g. *consume, consumption*) and/or by choosing only the FrameNet LUs which are in the same WordNet (Fellbaum, 1998; WordNet, 2006) synset or higher in the WN hierarchy than the original LU (e.g. *eat* to *consume*). Clearly this will exclude many well-formed paraphrases, so for tasks in which breadth is more important than accuracy of paraphrase, we anticipate experimenting with WordNet hierarchy distances between the original and paraphrase LUs as a quantitative measure of semantic similarity as a proxy for semantic well-formedness.

Currently, paraphrase scores are computed simply from the frequency of a particular valence in FrameNet data. We plan to significantly extend scoring to simultaneously rate each paraphrase on its WordNet similarity, syntactic edit distance[7], and language model scores. We also plan to measure the correlation between these estimated scores and both human-judged paraphrase accuracy and application dependent metrics, e.g. extension of in-domain language models by paraphrase.

WordNet can also be used to provide additional paraphrases beyond the particular valences attested in FrameNet. For example, we plan to use WordNet to generate synonyms of target words so that, for example, *adore* could be used anywhere *like* is used even if *adore* never appears in the FrameNet data.

Finally, the structure of the Mutaphrase algorithm makes multi-lingual paraphrase possible. This requires FrameNet-like data in other languages, and several projects are underway to provide just such a resource (FrameNet, 2007d; FrameNet, 2007c; SALSA, 2007). We plan to exploit these as they become available.

5 Conclusions

We have presented the Mutaphrase algorithm, a system for generating a large set of paraphrases of semantically marked input sentences using FrameNet. The generated sentences range widely in their similarity to the input sentence both in terms of syntax and semantics. Various methods of filtering the output for well-formedness and semantic and syntactic similarity were presented.

Although the current implementation suffers from a number of limitations, we believe these can be addressed, eventually providing a fully automated paraphrase system suitable for use in a variety of statistical natural language processing systems.

Acknowledgments

This work was partly supported by the European Union 6th FWP IST Integrated Project AMI (Augmented Multi-party Interaction, FP6-506811), and by the Swiss National Science Foundation through NCCR's IM2 project.

References

U. Baldewein, K. Erk, S. Padó, and D. Prescher. 2004. Semantic role labelling with similarity-based generalization using EM-based clustering. In R. Mihalcea and P. Edmonds, editors, *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 64–68, Barcelona, Spain, July. Association for Computational Linguistics.

C. Bannard and C. Callison-Burch. 2005. Paraphrasing with bilingual parallel corpora. In *Proceedings of the 43rd annual meeting of the Association for Computational Linguistics (ACL)*, pages 597–604, Ann Arbor, June.

R. Barzilay and L. Lee. 2003. Learning to paraphrase: An unsupervised approach using multiple-sequence

[5]It is worth noting that the current SemEval competition (FrameNet, 2007a) should lead to more complete automatic FrameNet-style annotation.

[6]An anecdotal example from a semantic parse of *I was prepared for a hound, but not for such a creature as this.* (Doyle, 1902) assigns *prepared* to the Cooking_creation frame, leading to the interesting paraphrase *I was tenderized for a hound....*

[7]We plan to base the syntactic distance on the edit distance between the original and paraphrase syntactic valences.

alignment. In *Proceedings of the Human Language Technology Conference (HLT)*, pages 16–23, Edmonton, Canada, May.

R. Barzilay and K. McKeown. 2001. Extracting paraphrases from a parallel corpus. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 50–57, Toulouse, July.

C. Callison-Burch, P. Koehn, and M. Osborne. 2006. Improved statistical machine translation using paraphrases. In *Proceedings of the Human Language Technology Conference (HLT)*, pages 17–24, New York City, June.

W. Dolan, C. Quirk, and C. Brockett. 2004. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING)*, Geneva, Switzerland, August.

A.C. Doyle. 1902. Hound of the Baskervilles. Project Gutenburg web site. http://www.gutenberg.org/dirs/etext02/bskrv11a.txt.

K. Erk and S. Padó. 2006. Shalmaneser — a flexible toolbox for semantic role assignment. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC)*, pages 527–532, Genoa, Italy, May.

C. Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. The MIT Press, May.

T. Fontenelle, editor. 2003. *International Journal of Lexicography Special Issue on FrameNet and Frame Semantics*. Oxford University Press, September. volume 16(3).

FrameNet. 2007a. The FrameNet task on SemEval web site. http://nlp.cs.swarthmore.edu/semeval/tasks/task19/summary.shtml.

FrameNet. 2007b. FrameNet web site. http://framenet.icsi.berkeley.edu.

Japanese FrameNet. 2007c. Japanese FrameNet web site. http://jfn.st.hc.keio.ac.jp/.

Spanish FrameNet. 2007d. Spanish FrameNet web site. http://gemini.uab.es:9080/SFNsite.

A. Ibrahim, B. Katz, and J. Lin. 2003. Extracting structural paraphrases from aligned monolingual corpora. In *Proceedings of the Second International Workshop on Paraphrasing*, pages 57–64, Sapporo, Japan, July.

D. Kauchak and R. Barzilay. 2006. Paraphrasing for automatic evaluation. In *Proceedings of the Human Language Technology Conference (HLT)*, pages 455–462, New York City, June.

R. Kittredge. 2002. Paraphrasing for condensation in journal abstracting. *Journal of Biomedical Informatics*, 35(4):265–277.

D. Lin and P. Pantel. 2001. Discovery of inference rules for question-answering. *Natural Language Engineering*, 7(4):343–360.

K. Litkowski. 2004. Senseval-3 task: Automatic labeling of semantic roles. In R. Mihalcea and P. Edmonds, editors, *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 9–12, Barcelona, Spain, July. Association for Computational Linguistics.

D. Moldovan, R. Gîrju, M. Olteanu, and O. Fortu. 2004. SVM classification of FrameNet semantic roles. In Rada Mihalcea and Phil Edmonds, editors, *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 167–170, Barcelona, Spain, July. Association for Computational Linguistics.

B. Pang, K. Knight, and D. Marcu. 2003. Syntax-based alignment of multiple translations: Extracting paraphrases and generating new sentences. In *Proceedings of the Human Language Technology Conference (HLT)*, pages 102–109, Edmonton, Canada, May.

C. Quirk, C. Brockett, and W. Dolan. 2004. Monolingual machine translation for paraphrase generation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 142–149, Barcelona Spain, July.

F. Rinaldi, J. Dowdall, K. Kaljurand, M. Hess, and D. Mollá. 2003. Exploiting paraphrases in a question answering system. In *Proceedings of the Second International Workshop on Paraphrasing*, pages 25–32, July.

SALSA. 2007. SALSA Project web site. http://www.coli.uni-saarland.de/projects/salsa/.

Y. Shinyama, S. Sekine, K. Sudo, and R. Grishman. 2002. Automatic paraphrase acquisition from news articles. In *Proceedings of Human Language Technology Conference (HLT)*, pages 40–46, San Diego, March.

WordNet. 2006. WordNet web site. http://wordnet.princeton.edu.

I. Zukerman and B. Raskutti. 2002. Lexical query paraphrasing for document retrieval. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING)*, pages 1–7, Taipei, Taiwan, August.