Table-driven Word Sense Disambiguation

Linas Vepstas

***Abstract*—The age-old observation that sense and syntax are correlated is used to build a table-driven word-sense disambiguation algorithm. Because the sense is determined by a table lookup, it is extremely fast, making it suitable for practical, real-time semantic-web applications. By contrast, existing word-sense disambiguation (WSD) or semantic role labelling (SRL) algorithms are fairly CPU-intensive, requiring many seconds or even minutes to run on modern hardware.**

The lookup table is constructed by correlating natural language parse structures with word-sense assignments. The key observation is that certain parse structures resemble very fine-grained part-of-speech (POS) tags, and that this fine-grained tag can be used to construct a more fine-grained (but otherwise traditional) word-sense lexicon. Manually creating such a dictionary is overwhelming, and so the lexicon is constructed using automated techniques: a large body of text is tagged with senses using a standard WSD algorithm, and then tabulated with the result of NLP parses of the same text. The POS distinctions are much finer than the usual noun/verb/adj/adv distinctions or even the Penn Treebank POS tags. These fine-grained tags allow the correlation between sense and syntax to be examined at a more detailed level than usual.

***Index Terms*—Lexicon, NLP, parsing, word-sense disambiguation, WSD, SRL, Semantic role labelling, Link Grammar**

I. Introduction

ATTENTION: READ THIS NOTE FIRST:

That there is a correlation between English-language syntax, and word senses, is obvious, and underlies the presentation of dictionaries: word senses are traditionally grouped according to part-of-speech (POS): noun, verb, adjective, adverb. Parts of speech are crude indicators of the allowed syntactic use of a word: so, in general, one cannot use a noun in place of a verb. Of course, there are exceptions: English grammar does allow the use of nouns as noun modifiers, the verbing of nouns by use of gerunds, and one can, to a limited extent, verbize one's nouns directly. But each of these grammatical operations alters the sense of a word as well as changing its POS category; the word sense is no longer the same, and most dictionaries supply distinct entries for each usage: senses and POS tags correlate.

Some dictionaries attempt a more fine-grained distinction of word senses: thus, for example, The American Heritage® Dictionary of the English Language[15] frequently uses section headings of *v.tr.* and *v.intr.* to denote word senses that are used only with the transitive or intranstive form of a verb. Thus, upon witnessing a verb used in a certain sentence in a certain way, one can immediately narrow down the possible senses for that verb.

It is is natural to presume that the correlation between word senses and word usage might continue to a finer, more detailed level, if only one could make finer, more detailed syntactic observations, and had a finer, more detailed lexicon to work with. This is the primary thesis of this paper: such finer distinctions are possible and useful. To illustrate this, consider, for example, the verb "*to suffer*". The American Heritage Dictionary lists seven senses for this; other dictionaries list from 5 to 11 senses. The Princeton WordNet dictionary[2] lists 11 meanings. Now consider the sentence:

"She suffered a fracture in the accident."

A syntactic analysis shows that "*suffer*" is in the past tense, takes a singular pronomial subject, a singular direct object, and a modifying prepositional phrase. The corresponding WordNet sense key is *suffer%2:29:01::, (undergo (as of injuries and illnesses))*. This example is syntactically quite different than those of other example sentences taken from WordNet:

"This author really suffers in translation."

where the verb is in the present tense, has a singular subject, a prepositional modifier, but no object: it corresponds to sense *suffer%2:30:02::, (be set at a disadvantage)*. Similarly one has

"Many saints suffered martyrdom."

which has a plural subject, is in past tense, and the object is an uncountable (mass) noun: this corresponds to the sense *suffer%2:39:01::, (undergo or be subjected to)*. Each of these example sentences, taken from WordNet itself, shows a strong correlation between the syntactic structure of the sentence, and the designated word sense.

Most of this correlation can be erased by altering the example sentences, and creating new, grammatically correct sentences, by altering the tense, the count of the subject or object, the presence or absence of the object, or adding/removing a modifying perpositional phrase. But do native English speakers generate all possible sentence constructions with equal probability? Clearly not: idioms, the strict coupling of a few words, are a well-known liguistic phenomenon. This observation can be taken much farther: the Mel'cuk Meaning-Text Theory[6], [16] posits that the intended meaning of an expression strongly influences the structure of the sentence that expresses that meaning; conversely, that certain sentence patterns are used only with limited, well-defined subsets of nouns, verbs. Thus, the proper study of the correlation between sense and syntax, in order to be valid and correct, must become an exercise in corpus linguistics: the tabulation of the frequency of word usage as used by actual writers, as compared to the intended sense of the author. To the extent that such a correlation exists, it can be tabulated into a dictionary, and this dictionary can, in turn, be used to quickly guess at the intended meaning, given only the syntactic usage.

Discovering this correlation relies on large-scale computational linguistics techniques. Although one might be able

Linas Vepstas is with Novamente LLC; work was supported by Cerego/smart.fm

to analyze, by hand, a few verbs or nouns, in the manner illustrated above, this is hardly practical to get a true sense of the prevelance of this correlation. In doing a small-scale analysis, one might discover only a few words that appear to be highly correlated, without being able to say much about the language, in general. What is wanted is a full lexicon that makes fine-grained POS distinctions. Its not practical to build such a lexicon by hand; it must be constructed in some automated way. The approach used here is to parse a large quantity of text, obtaining syntactic structure, and to tag the same text with word-senses, and so as to generate statistics correlating the two tag sets.

The "weak link" to this automated approach is sense tagging. The quantity of text hand-tagged by experts with word senses is not large enough to usefule for collecting a reasonable amount of statistics; thus sense-tagging must be performed automatically. Although automated sense tagging or "word sense disambiguation" (WSD) algorithms have been improving, the best simple, straight-forward systems remain marginal in precision and recall. This raises an interesting question: is it possible that, by correlating sense tags with syntactic usage, that the incorrect tags will statistically "cancel out", while correct tags will reinforce? That is, is it possible that a lookup-table based WSD algorithm might have euqal or better accuracy than the underlying WSD system from which it was constructed?

For parsing, the Link Grammar parser[13], [14] is used. The output of the Link Grammar parser are "linkages" or "disjuncts" that explicitly express how a word is connected to its vicinity. Each disjunct is essentially a linkage statement, such as "this is a verb with a singular subject on the left, a singular direct object on the right, and a modifying prepositional phrase", expressed in a compact notation ("*Ss- Os+ MVp+*" for this example). These disjuncts are taken as the "fine-grained POS tags" refered to above. In certain ways, Link Grammar resembles dependency parsing, and indeed, it is straight-forward to convert a Link Grammar parse into a dependency parse[1]. Unlike a dependency parse, Link Grammar does not explicitly indicate head words; but this is not required for the notion of fine-grained POS tags. Conversely, it is straight-forward to create a "fine-grained POS tag" from a dependency parse; thus, the results in this paper should be reproducible by employing other parsers with dependency output. It is less clear how to extract a "fine-grained POS" from a phrasal (constituency, or "phrase structure") parse: phrasal parses do not indicate the relationships and constraints between words; rather, the production rules of such parsers contain many non-terminal (non-word) symbols. Perhaps the sequence of production rules taken to arrive at a given word could serve to act as a "fine-grained POS tag" for that word. Understanding how different parsers change the the correlation between POS and sense tags may shed light into the nature of the syntactic/semantic connection. This question is not explored in this paper.

[1]The RelEx semantic relationship extractor[[XXX need ref] accepts Link-Grammar parses as input, and, among other outputs, can generate dependency parses fully compatible with the well-known Stanford dependency parser.

Word-sense tagging was performed using an implementation of the Mihalcea all-words word-sense disambiguation algorithm[9], [8], [12]. This algorithm is reasonably accurate and is straightforward to implement. It requires the use of a word-sense similarity measure; several such measures available for WordNet. The output of the Mihalcea WSD is a ranked list of word-sense assignments (WordNet senses) for each word in a text. The final dicitionary created is then a tabulation of the frequency with which a given Link-Grammar disjunct was observed with a given WordNet sense key.

It is important to recognize that table-driven patten recognition is a biologically plausible model for human cognition. Pattern recognition is a central precept of the connectionist philosophy of human cognition [xxx need ref]. It seems unlikely that the human brain implements proceedural computational algorithms; rather, connectionism states that symbol manipulation is performed with neural-net style architectures. A table-lookup based pattern recognizer fits neatly into this picture: one might imagine a single neuron with a set of connections sensitive to a particular syntactic pattern; when that pattern is presented on the input, the neuron fires. In this sense, the table lookup of a pattern resembles a single-layer discriminatory neural net or perceptron; for a given input, only a small number of outputs (word senses) are suggested. This analogy indicates the power as well as the weakness of the tehnique. Table lookup can be made massively parallel: those neurons that know of a specific pattern respond, and all others are silent. This echoes the massively-parallel structure of the brain. But single-layer perceptrons are also notoriously limited: they can classify "linerarly seperable" patterns, but no more. There is no doubt that language is far more complex than that. There is no reason to believe that table-driven WSD could ever function more accurately than a single-layer perceptron: its strength is speed, and a certain degree of "biological naturalness". One does not expect exquisite accuracy. In the connectionist philosophy, greater sophistication requires that the output of one layer be fed to the input of another: thus, for example, table-driven WSD might provide a-priori weighted word-sense suggestions for other, more refined semantic algorithms.

The mechanism used to construct the lexicon is also fairly connectionist in nature. The Mihalcea all-words WSD graph algorithm resembles a Markov chain, solved using the PageRank algorithm, and is thus fairly explicitly connectionist in and of itself. The Link-Grammar parser itself is currently implemented using a computational, not connectionist algorithm. However, the author beleives that it should be possible to re-implement the parser using the Viterbi algorithm. This could provide a significant speedup, especially for long sentences, as well as making the parser more biologically natural: the Viterbi algorithm is explicitly connectionist, maintaining only a finite history, with a finite number of connections to recent input (i.e. enouraging and maintaining mosly just short-distance connections between words).

Breif results summary ...

Outline of paper....

use "semantic role labelling" in a few more paras.

– xxx also make the following important point: the semeval

code (mihalcea 1994) indicates that the "most gfrequent sense" – i.e. MFS provides a very strong signal. So the work here is all about how disjuncts modify the MFS signal!

II. Previous Work

Perhaps the most developed of linguistic theories that attempt to describe how syntax gives rise to semantics is Igor Mel'čuk's "Meaning-Text Theory"[16], [6]. The theory describes a network of relationships between semantic lexemes and a set of functions and proceedures that relate this network to corresponding grammatically correct utterances. In particular, the theory attempts to explain why a "phraseme" or "set phrase", such as the phrase "*to give X a look*", constricts the kind of things that can occupy the slot *X*. In this example, *X* typically names a process, a product, an idea or proposal. A key observation in MTT is that, semantically, *X* cannot be just "any old noun"; even though pure syntactic analysis would allow this. To fill this slot in a semantically meaningful way, a noun would have to name or denote something worth pondering, reviewing, looking over. One of the acheivements of the theory is to precisely specify how such a restriction comes about and how it can be maintained in a computational way.

The structure of an MTT "Explanatory Combinatory Dictionary" is quite different than an ordinary dictinary: it gives precise and detailed instructions on how to convert lexical meaning into syntactic expression. In this sense, it correlates syntax and semantics. By contrast, the work described in this paper shows how to automatically create a short-cut or abridged form of such an ECD: for any given word-sense, one has a table of allowed syntactic structures in which that word sense has been observed (and a frequency count of how often that word-sense was used in this particular way). The MTT ECD is hand-constructed by linguists after hard, patient work. The approach described here obtains relationships automatically, with an unsupervised learning algorithm.

There are other approaches for the unsupervised learning of syntactic/semantic correlations. These include Dekang Lin's work on automatic synonym discovery[3], [4], the work of Domingos and others on applying Markov Logic Networks (MLN)[1] to semantics[11], [7]. Each of these approaches also have one strength that the current approach does not: they potentially discover much narrower classes of words that can appear in juxtaposition to each other. That is, the current Link Grammar word classes are, for the most part, broad, syntactic categories, and not narrow, semantic classes. So, for example, the Link Grammar link *Os+*, which is short-hand for the statement "*this word takes a singular direct object on the right*", places no particular restriction on what that object might be – it can be any noun. This is in contrast to the "*give X a look*" example, where X is syntactically free, but semantically constrained.

Lin describes how to automatically build a thesaurus in [3], by examing the frequency with which words are used in similar dependency relations. Specifically, he creates a dependency parse, having relations of the form $r(w_1, w_2)$ with r a relation (such as *subj*, *obj*, *etc.*), and w_1 and w_2 are words. By examing co-occurance statistics for such relations in a large corpus, he provides a number of similarity measures which are able to successfully identify synonymous words. This result is quite remarkable. It can be criticized in two ways: by collecting statistics on just individual relations, it does not look at the broader syntactic context of a word. That is, for example, the meaning of a word may depend on its having both a direct object, and a prepositional relation, both within the same sentence. This is partly corrected in a later work[4], which groups together dependency paths to discover synonymmous phrases. A similar solution is explored in the Markov logic work described below. Either approach provides for the automatic discovery of significant parts of an entry of the ECD dictionary of MTT theory. A second criticism is that the work does not suggest how to discover that a given word may have multiple meanings: it does not specify how to group synonyms into synonym sets. As such, there is no direct way to extend this work to perform sense labelling on individual words in a sentence: there is no way to look at a word in a sentence and discover the intended lexeme.

A global approach to automatic synonymous phrase discovery is offered by applying the theory of Markov logic networks to the extraction of semantic content from a dependency parse[11]. The approach used extracts, tabulates and clusters sentence patterns according to common substructures, thus automatically discovering synonymous expressions by performing a global sentence analysis. In certain ways, it can be thought of as a generalization of Lin's DIRT[4]. The system is remarkable in that it is completely unsupervised (does not require a training corpus), and requires very few *a priori* built in rules. Although it significantly outperforms other question answering systems, this result can also be taken as a critique: it does not explicitly provide markup or tagged output; but rather performs question answering by pattern matching.

A contrasting application of Markov logic nets is that of Meza-Ruiz, *et al*[7]. This system uses MLN to simultaneously provide a variety of tags, and, in particular, to identify the head-word of a sentence, disambiguate the semantic frame that it participates in, and provide a semantic role tag (essentially, a word-sense) for the head-word. Unfortunately, the system uses a large number of hand-crafted relations, frames and rules to specify the logic network. However, once trained, the weighted network can be quickly evaluated to obtain tags for any given sentence, typically in fractions of a second for contemporary computers.

It is an interesting exercise to contrast the above-descibed global sentence analysis systems to the current work in dependency parsing, such as minimum spanning tree approaches[5] or greedy state machines[10], where global structure is essentially ignored, except to constrain the parse to include all of the words in a sentence. In essence, syntax is a local constraint on word juxtapositions; semantics is a global anlysis of word relationships. This fits well into the theoretical framework of Meaning-Text Theory; that the current work lies in a middle ground is perhaps no accident.

XXXXX

More prev work:

Lexical semantics: word-sense selection constraints are en-

coded in individual lexicon entries using syntactic dependencies or other forms of "licensing" or "expectations." Systems that take this approach include that of Wilks (1975) and other conceptual analyzers such as CA (Birnbaum and Selfridge, 1981) and Word Expert Parser (Small and Rieger, 1982).

Wilks, Y. 1975. A preferential, pattern-seeking, semantics for natural language inference. Artificial Intelligence 6(l):53-74.

Birnbaum, L. and Selfridge, M. 1981. Conceptual analysis of natural language. In Inside Com- puter Understanding, ed. R. Schank and C. Riesbeck, p. 318-353. Lawrence Erlbaum Associates.

Small, S. L. and Rieger, C. 1982. Parsing and comprehending with word experts. In Strategies for natural language processing, ed. W. G. Lehnert and M. H. Ringle. Lawrence Erlbaum.

III. Methodology

This section provides a breif review of the ingredients for creating the table of POS-sense tags: the Link Grammar parser, the Mihalcea WSD algorithm.

The theory of Link Grammar[13], [14] is built upon the concept that allowed "connections" between words are determined by a set of "connectors" attached to the words. Words with connectors on them can be thought of being like puzzle pieces: they can only be correctly assembled when mating pieces match. The link-grammar lexicon contains lists of words together with the connectors that they carry. Consider then the sentence "*Mary loves beer*". The Link Grammar dictionary might contain (for example) entires such as these:

```
beer: O-
loves: S- & O+
Mary: S+
```

Here, `O-` and `S+` are "connectors", and `S` and `O` are the "link types" of the connectors. The above is simplified for illustration: in practice, words can carry a large variety of different connectors. The example sentence, as parsed by the Link Grammar parser, it is rendered as

```
+--S--+--O--+
|     |     |
Mary loves beer
```

This parse is obtained by noting that `S` connectors can connect only to other `S` connectors, and, specifically `+` denotes a connection to the right, and `-` a connection to the left. Thus, `S+` can only connect to `S-` to form an `S` link; likewise for `O+` and `O-`. An expression such as `S+ & O-` will be called a "disjunct" in the remainder of the text; this is because such an expression is usually disjoined with many others that a word can have. The expression `S+ & O-` implicitly encodes a verb-like connection pattern for the word "*loves*" – it demands a subject on the left, and an object on the right.

The actual Link Grammar dictionary is only slightly more complicated than what the above suggests. In practice, there are over 100 different link types. Many link-types also have subtypes, which are optionally matched using a form of wild-card matching. Some connectors, such as those to modifiers, are optional. Thus, a realistic example from the current version of the parser:

```
                +------------Ou------------+
   +-----Ss-----+        +--------A---------+
   |    +--Em--+        +-Xc-+    +----A---+
   |    |      |        |    |    |        |
Mary really loves.v dark.a , heavy.a beer.n
```

Here, the "subscripts" `.f`, `.v`, `.a`, `.n` are non-functional tags that help simplify the organization of the dictionary, although they also hint at the word function: `.v` for verbs, `.n` for nouns, *etc*. The link type `Ss` indicates that the subject is singular, and `Ou` that the object is uncountable. The dictionary entry `loves.v` consists of 40 different disjuncts, some of which are:

```
({@E-} & Ss- & O+) or
({@E-} & Ss- & TO+) or
({@E-} & Ss- & Pg+)
```

Here, `{@E-}` indicates that "*loves*" can take one or more optional emphasis modifiers, while `TO+` would link to the word "*to*" ("*Mary loves to dance*"), and `Pg+` would link to gerunds ("*Mary loves dancing*"). The disjunction `or` indicates that only one of these disjuncts will be used in a parse.

Upon completion of parsing, we see that the word "*loves*" was de facto assigned only one disjunct: namely `Em- Ss- Ou+`. Here, we drop the ampersand, for brevity; it is implicit. This disjunct is is what will play the role of the "fine-grained part of speech" in the statistical tabulations of parsed text.

Word-sense tagging was performed by a simple implementation of the Mihalcea WSD algorithm[8], [12]. In this implementation, every word in a sentence is tagged with all possible word-senses appropriate for the assigned coarse part-of-speech, taken from the WordNet 3.0 dictionary (WordNet defines only four parts of speech: noun, verb, adjective and adverb). A clique graph is created, by taking each sense to be a vertex, and with edges running between all possible sense pairs. The edges are assigned numeric weights, based on the similarity of the word-senses at each end of the edge. Word-sense similarity was computed using the perl Sense::Similarity package [xxx need ref], and using the xx similarity measure between verbs, and the xx measure between nouns, as suggested in [12] to obtain the best scores. Because these two similarity measures run over a different set of numeric ranges, they must be rescaled to be directly comparable. The weights were renormalized so as to xxx. The clique can be thought of as a matrix, which can be renormalized to define a Markov chain. One common way to find an approximate solution to such a Markov chain is the employ the PageRank algorithm. The result of doing so is a vector of probabilities, assigning a score to each possible word sense. The algorithm works due to a simple observation: word-senses used in a sentence tend to be closely related; the use of a Markov chain spreads around this notion of "relatedness" in an equitable manner. The end result is a ranked, weighted list of WordNet 3.0 senses for each word. Taken together with the disjunct tag obtained from parsing the sentence, one can now tabulate statistics.

Tabulation was performed by simple counting, accumulating results into an SQL database storing (word, word-sense, dis-

junct) triples, weighting the count by parse confidence and by the assigned word-sense confidence. The Link Grammar parser will produce multiple parses for a sentence when it finds the grammatical structure of the sentence to be ambiguous. It will rank these parses according to a confidence measure considting of four integers: linkage-length, disjunct-cost, and-cost, unsed-word cost. The linkage-length simply counts the total length of the links in a parse. The disjunct-cost is incurred when a "costly" disjunct is used to obtain a parse. The and-cost refers to imbalanced branches of sentences in conjunctions; unused words can occur when the parser cannot comprehend a sentence. These four numeric scores are combined into a single floating-point parse-ranking score for the sentence with the *ad-hoc* formula:

$$\begin{aligned}\text{SCORE} = \quad \exp \quad & -(0.012 * \text{LEN} + 0.06 * \text{DIS} + \\ & 0.2 * \text{AND} + 0.4 * \text{UNUSED})\end{aligned}$$

This formula always gives a value of less than 1.0; it typically assigns scores in the range of 0.9 for the confident parses of shorter sentences, and can give scores as low as 0.1 or less for parses that were problematic. If there was more than one parse for a sentence, all of the highest-ranked parses were considered, up to a total of four. For each alternative parse, this parse score was used as a multiplier to weight the word-sense score. These weighted scores were accumulated into the (word, word-sense, disjunct) triples table.

Notice several ad-hoc assumptions – these adhocs should be explored:

– parse ranking formula

– re-ranking of word-sense scores.

IV. Corpus

IMPORTANT: READ THIS NOTE:

The corpus that was analyzed for this work consists of about 41 thousand (41K) articles starting with the letters A-E from a May 2008 dump of the English-language Wikipedia. On average, there were 22 sentences per article, 22 words per sentence, and 2.8 distinct parses per sentence. Including multiplicities for the multiple parses, a total of 58 million words were observed. A total of 230K distinct words were observed, of which 117K were observed at least four times. This number is large for several reasons: the English language Wikipedia includes many foreign words, as well as a large number of names of entities (people, places, *etc*.). In addition, the parser identifies numbers (times, dates, measurements) as unique words, thus further inflating the raw word count. By comparison, the Link Grammar parse dictionaries contain 76K words, counting inflected forms (plurals, tenses, *etc*.). After lemmatization, which is required in order for a word to be found in WordNet, only 15K distinct word lemmas were observed. By comparison, WordNet contains 155K entries, of which 117K are nouns, many of these being entity names or noun phrases.

These words were tagged with 46.5K unique, distinct disjuncts. These wre composed of 385 unique link types, not counting idiom links. These 385 links are subtypes of the 89 primary link types; as explained above, subtypes are typically used to enforce number or tense agreement, or other types of agreement restrictions. Notable is tht Link Grammar defines 103 different major link types; not all link types were used in this corpus! The unused types deal primarily with questions, some fairly idiomaitc comparative usages, and certain types of numeric expressions involving fractions. Perhaps this is not surising: one doesn't expect questions or judgemental statements in Wikipedia articles. There were 45 idiom links observed, again a surprise, as the Link Grammar dictionaries contain considerably more idiomatic phrases. Idiom links are unique, automatically-generated link types to used to connect the words in an idiomatic phrase; they have no particular significance, other than to maintain a consistent approach to the implementation of the parser.

A disjunct is then some combination of these 385+45=430 different link types, together with a +/- direction indicator on each, to form a connector. Disjuncts may contain repeated connectors; thus, for example, if a noun had appeared with two adjective modifiers to its left, it will include "`A- A-`" in its disjunct to signify these two connections. So for example, "`A- A- Ss+`" indicates a noun with two adjectival modifiers, used as the subject of a sentence. Ignoring disjuncts containing idiom connectors, a total of 46.1K unique disjuncts were observed – a tad less than before, but indicating that idiomatic phrases make a minimal impact on the total number of possible word connections. In total, 227K unique word-disjunct pairs were observed; however, many of these were observed only infrequently; only 56.6K were observed at least five times.

The words were tagged with 19.7K unique different word senses. This is only a small fraction of the 207K senses contained in WordNet 3.0. Curiously, there were fewer senses identified and used than there were different grammatical usages (*i.e*. disjuncts). There were a total of 568K (word,disjunct,sense) triples observed. Again, many of these were seen very infrequently: only 76K triples were observed more than five times.

V. Results

XX Need intro?

A basic premise of this paper is that correlations between syntactic use and word-sense exist. This section quickly presents a few examples that were discovered that sustain this view.

Given a table of (word, disjunct, sense) triples, with frequency counts, there are several quantities of interest. Define $P(w, d, s)$ to be the probability of observing the (word, disjunct, sense) triple (w, d, s). The conditional probability

$$P(s|w, d) = \frac{P(w, d, s)}{P(w, d, *)}$$

gives the probility of seeing the sense s given a fixed w, d. If this is close to 1, this indicates that a a certain word sense is being used almost uniquly/exclusively in a certain syntactic context. By contrast, if this is close to zero, this indicates that the given sense is almost never used with the disjunct.

An alternative way of observing relationships between sense and syntax is by means of the entropy

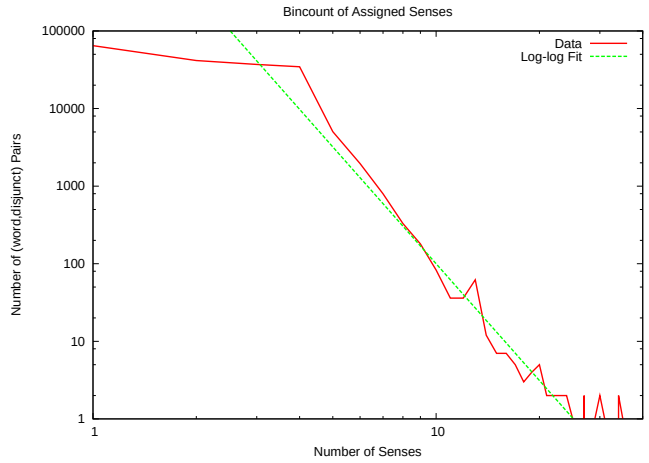
$$H(w,d) = -\sum_s P(s|w,d)\ \log_2 P(s|w,d)$$

For the case where there is only one sense s observed associated with w,d, one has $P(s|w,d) = 1$ and so $H(w,d) = 0$. In general, if there are only a small number of senses that are used with given disjunct-word pair, or if one sense dominates over all others, then the value of $H(w,d)$ will be small. Low-entropy disjunct-word pairs indicate a strong association of sense and syntax.

Consider, for example, the verb '*suffered*', together with the disjunct '`Ss- Os+ MVp+`', which indicated that 'suffered' was used with a singular subject, a singular direct object and a prepositional modifier. This word, disjunct pair was observed 176 times. It was tagged with eight different senses (out of 11 possible in WordNet). It was tagged wtih the sense *suffer%2:29:01::* in 138 of those uses, or 138/176=78% of the time. The total entropy for this triple was 1.274, which is considerably higher than the average of 0.777 for the entire dataset. The average number of senses observed per (w,d) pair was 2.373, but the median was just 1: of the 227K total observed (w,d) pairs, 89K, or 39%, were assigned just one sense.

The distribution of sense assignments is shown in figure1. This figure just shows a histogram or bincount: how many of the (w,d) pairs were assigned a given number of senses.

Figure 1. Distribution of word-sense assignments



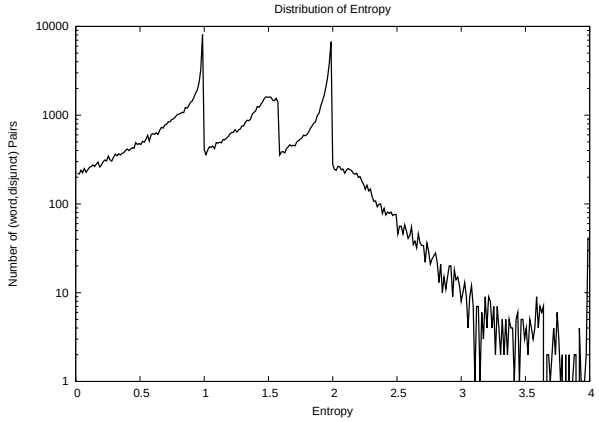
The figure above shows the distribution of (word,disjunct) pairs with respect to the number of associated senses assigned to the pair. As obvious fom the "plateau" in the graph, almost all (word,disjunct) pairs were assigned 4 or fewer senses. The drop-off for five or more senses is very steep, with the straight line marked "log-log fit" being given by $10^7 s^{-5}$, where s is the number of senses.

The distribution of entropy vs. (w,d) is shown in the figure below.

This was observe

– what is the total entropy? what would max ent be?

Figure 2. Distribution of Entropy



Note the prominent peaks in the graph above. This is due to a bug in the tagging algo: where there were islands in the page rank algo (i.e. regions disconnected from the main graph) these islands weren't properly tagged, and instead, senses were given equal probability assignments: i.e. 1/2, 1/2, or 1/3,1/3,1/3 etc. which gives rise to large, prominent peaks in the entropy graph. The sharp drop to one side, and decay to the other, is easily explained: in many of these cases, there are small tie-breaker votes from other sources. These tie-brakers skew the totals to be e.g. 51%-49%, which has a slightly lower entropy than 50-50. Argh !!!!!

The above suggests that the entire dataset should be completely redone.

– how many have low ent?

–what about results by POS category?? e.g. does this work better for verbs, or for nouns? wahat about adj/adv? (i.e. accuracy?)

Total count is 593003.630924276 for 537985 items Done updating the probs Done updating the entropy of 226679 disjuncts Avg sense cnt=2.37333409799761 avg entropy=0.777430272279844;

A single column figure goes here

Figure 3. Captions go *under* the figure

Table I
TABLE CAPTIONS GO *above* THE TABLE

delete	this
example	table

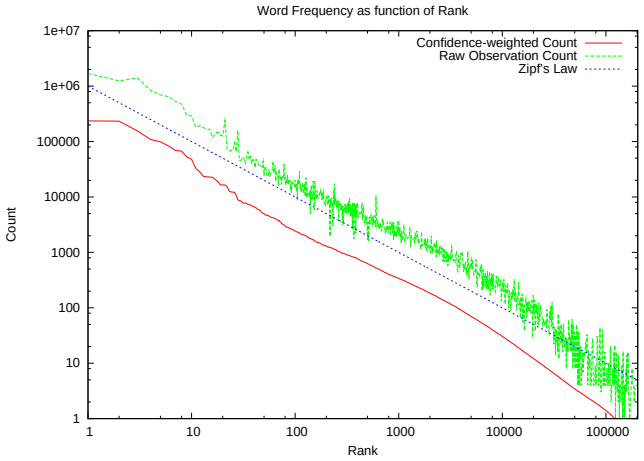
VI. DO IT AGAIN

The stuf here from the "new" tables of Dec 2009/Jan 2010

VII. CONCLUSIONS

One of the primary applications of the WSD effort is to provide the Link Grammar parser with a set of dictionaries theat provide parse-ranking, and provide parse-time

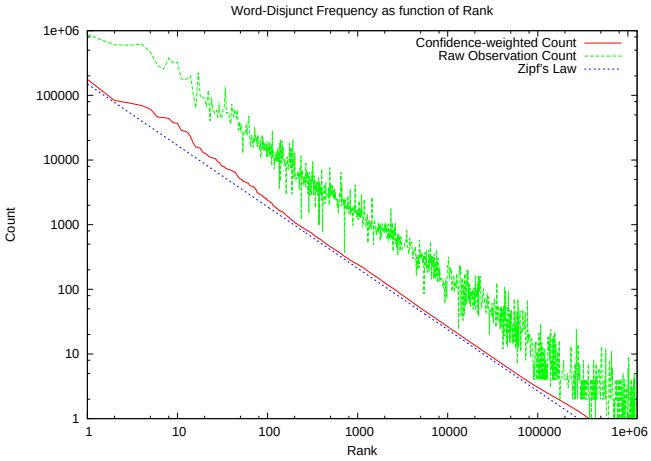
Figure 4. Distribution of POS-tagged Words



This figure shows the distributation of POS-taggged words in the analyzed corpus. The parser tagged each word with a part of speech; categories include noun, verb, adjective, adverb, although over a dozen other miscellaneous tags are also used. These tagged words were counted in two diferent ways: a "raw" observational count, of 1, for each occurance, and a weighted count, where the entire parse is assigned a parse-confidence score in the range of 0.0 to 1.0; the word is then counted with the parse-confidence score.

To generate the graph, the words were ranked according to thier parse-weighted count, then this count, and the raw observational count, were graphed. The dashed "Zipf's law" line shows the graph $10^6 r^{-1.0}$ with r the rank. A total of 27.093 million tagged word occurances were observed, for a total weight of 3.821 million. A total of 202899 distinct tagged words were seen. This total does not include words with a parse-rank of 0.0; a relatively small number of these occur due to several effects, including bad pre-processing which failed to strip out html markup, footnotes and other typsetting markup in the input text.

Figure 5. Distribution of Word-Disjunct Pairs



This figure shows the distribution of word-disjunct pairs, as a function of rank. This figure is generated in the same way as the previous one; only the dataset differs. There were 129093 distinct disjuncts that were paired with various words during parsing. In total, there were 1.264 million distinct word-disjunct pairs observed. Since there are 202899 distinct tagged words in this dataset, this implies that each word was used with an average of 1264K/203K=6.2 different disjuncts.

The dashed "Zipf's law" line shows the graph $1.5 \times 10^5 r^{-0.95}$ with r the rank. Note that this exponent is rather unusual for Zipf's law; it is more common to observe distributions that turn downward more sharply (i.e. have an exponent such as -1.05), rather than less so. It is not clear if this curious slope is an artifact of the parse dictionary, or whether this is somehow a reflection of the behaviour of the English language itself.

syntactical word-sense disambiguation. See the file renorm-stats/README for instructions on how to prepare these files.

other conclusions

References

[1] Pedro Domingos, Stanley Kok, Hoifung Poon, Matthew Richardson, and Parag Singla. Unifying logical and statistical ai. In *AAAI'06: Proceedings of the 21st national conference on Artificial intelligence*, pages 2–7. AAAI Press, 2006.

[2] Christiane Fellbaum, editor. *WordNet: an electronic lexical database*. MIT Press, 1998.

[3] Dekang Lin. Automatic retrieval and clustering of similar words. In *Proceedings of the 17th international conference on Computational linguistics*, pages 768–774, Morristown, NJ, USA, 1998. Association for Computational Linguistics.

[4] Dekang Lin and Patrick Pantel. Dirt: Discovery of inference rules from text. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'01)*, pages 323–328. ACM Press, 2001.

[5] Ryan McDonald, Kevin Lerman, and Fernando Pereira. Multilingual dependency analysis with a two-stage discriminative parser. In *CoNLL-X '06: Proceedings of the Tenth Conference on Computational Natural Language Learning*, pages 216–220, Morristown, NJ, USA, 2006. Association for Computational Linguistics.

[6] Igor A. Mel'cuk and Alain Polguere. A formal lexicon in meaning-text theory. *Computational Linguistics*, 13:261–275, 1987.

[7] Ivan Meza-Ruiz and Sebastian Riedel. Jointly identifying predicates, arguments and senses using markov logic. In *NAACL '09: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 155–163, Morristown, NJ, USA, 2009. Association for Computational Linguistics.

[8] Rada Mihalcea. Unsupervised large-vocabulary word sense disambiguation with graph-based algorithms for sequence data labeling. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 411–418, Morristown, NJ, USA, 2005. Association for Computational Linguistics.

[9] Rada Mihalcea, Paul Tarau, and Elizabeth Figa. Pagerank on semantic networks, with application to word sense disambiguation. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, Morristown, NJ, USA, 2004. Association for Computational Linguistics.

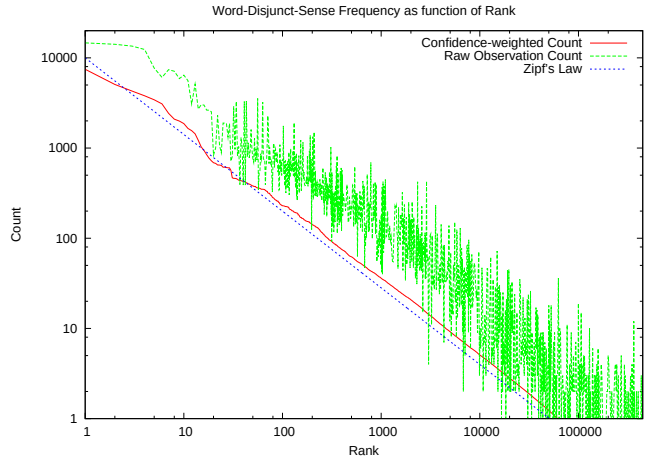
[10] Joakim Nivre. *Inductive Dependency Parsing (Text, Speech and Language Technology)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[11] Hoifung Poon and Pedro Domingos. Unsupervised semantic parsing. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1–10, Singapore, August 2009. Association for Computational Linguistics.

[12] Ravi Sinha and Rada Mihalcea. Unsupervised graph-basedword sense disambiguation using measures of word semantic similarity. In *ICSC '07: Proceedings of the International Conference on Semantic Computing*, pages 363–369, Washington, DC, USA, 2007. IEEE Computer Society.

[13] Daniel Sleator and Davy Temperley. Parsing english with a link grammar. Technical report, Carnegie Mellon University Computer

Figure 6. Distribution of Word-Disjunct-Sense Triples



The figure above is analogous to the previous two, showing the distribution of word-disjunct-sense triples. A total of 447672 distinct triples were observed 2.290 million times, and were assigned a cumulative confidence score of 433711. The confidence score is a product of the parse-confidence score for the word-disjunct pair, and a score for the confidence of the word-sense assignment. The dataset contains only 15771 distinct words, which were tagged by 15295 distinct senses, or slightly less than 1 sense per word. This is possible, as many words can be synonyms of one-another, even when they may have multiple senses. The dataset has 186490 distinct word-disjunct pairs, thus an average of 447672/186490=2.40 different word senses were assigned to each word-disjunct pair. This average, more than the last one, suggests that, indeed, gramatical usage can differentiate between senses.

The dashed "Zipf's law" line shows the graph $10^4 r^{-0.85}$ with r the rank. This exponent is even flatter than the previous one; again, it is not clear if this is an artifact of the parsing and sense-tagging algorithms, or a feature of the English language itself.

Science technical report CMU-CS-91-196, 1991.

[14] Daniel D. Sleator and Davy Temperley. Parsing english with a link grammar. In *Proc. Third International Workshop on Parsing Technologies*, pages 277–292, 1993.

[15] American Heritage Staff. *The American HeritageÂő Dictionary of the English Language*. Houghton Mifflin Company, fourth edition edition, 2000.

[16] James Steele, editor. *Meaning-Text Theory: Linguistics, Lexicography, and Implications*. University of Ottowa Press, 1990.

PLACE PHOTO HERE

Linas Vepstas All about me. I do many things, like NLP, math, programming,etc.