WASP-Bench: an MT Lexicographers' Workstation Supporting State-of-the-art Lexical Disambiguation

Adam Kilgarriff, David Tugwell

Information Technology Research Institute (ITRI), University of Brighton
Lewes Road, Brighton, UK
{adam.kilgarriff, david.tugwell}@itri.brighton.ac.uk

Abstract

Most MT lexicography is devoted to developing rules of the kind, "in context C, translate source-language word S as target-language word T". Very many such rules are required, producing them is laborious, and MT companies standardly spend large sums on it. We present the WASP-Bench, a lexicographer's workstation for the rapid and semi-automatic development of such rule-sets. The WASP-Bench makes use of a large source-language corpus and state-of-the-art techniques for Word Sense Disambiguation. We show that the WSD accuracy is on a par with the best results published to date, with the advantage that the WASP-Bench, unlike other high-performance systems, does not require a sense-disambiguated training corpus as input. The WASP-Bench is designed to fit readily with MT companies' working practices, as it may be used for as many or as few source language words as present disambiguation problems for a given target.

Keywords

MT-lexicography, Lexical Disambiguation, workbench.

Introduction

Choosing the right target language word is the central problem of MT. The difficulty usually lies in the fact that the source language word has different meanings, which translate differently, or that it translates differently in different contexts. Most MT lexicons have extensive sets of rules which say "for source-word *s*, if the context is *c*, translate as target-word *t*". The problem with these rules is that they are laborious to write, and you need a lot of them. In this paper we present a corpus-based lexicographers' workstation, in which this process is automatically supported. The software finds, from a large source-language corpus, a candidate set of contexts *C* which appear salient for a source word, so are good candidates for the left hand side of

context is $c \Rightarrow$ translate as t

rules. The lexicographer provides the appropriate value of *t* for a few of these, in a process that takes just a few minutes per source word. The software then applies the highest-performing Word Sense Disambiguation (WSD) algorithm currently available (Yarowsky (1995)) to bootstrap a long list of rules from the 'seeds' the lexicographer has provided. The list is ordered by confidence: the rules at the top of the list are the ones we can be surest of. Thus to disambiguate, and thereby select the appropriate target word *t*, for an instance of *s* in a text to be translated, all an MT system need do is work through the list until it finds a rule where the left hand side matches. It can then select the target-language word that the rule specifies.

The WASP-Bench is designed to fit readily with MT companies' working practices. It can be used for as many or as few source language words as present disambiguation problems for a given target, making it suitable for lexicon maintenance as well as development. The rule-sets, or "word sense profiles", comprise simple, surface-context-based rules and should be straightforward to integrate into most MT software. For some languages, large source-language corpora will not be available, but with the increasing availability of electronic versions of newspapers and large volumes of text on the web (Grefenstette & Nioche, 1999) for scores of languages, this is ceasing to be an issue for many languages.

Below, we first situate the WASP-Bench with respect to the lexicography and WSD literatures; we then describe how the WASP-Bench works, and how the lexicographer interacts with it; then, an evaluation, using the English SENSEVAL dataset for WSD evaluation (Kilgarriff & Palmer, 2000). We show that the WSD accuracy is on a par with the best results published to date, with the advantage that the WASP-Bench, unlike other world-leading systems, does not require a sense-disambiguated training corpus as input.

Word Sense Disambiguation

Word Sense Disambiguation has developed as a sub-discipline of computational linguistics largely separately from machine translation activity, even though the one arena in which WSD is definitely required, and can immediately be put to use, is MT (Kilgarriff (1997)). Within the MT community, the task is more often called "lexical disambiguation" (in this paper we shall use "WSD"). There has been great progress in WSD over the last ten years, with the best results emerging from the application of machine learning technologies to large text corpora (Ide & Veronis, 1998).

As systems got better, and there were more of them, so issues of evaluation came to the fore. In response, the SENSEVAL evaluation exercise (Kilgarriff & Palmer, 2000) was set up. For a small sample of 40--50 words, a set of around 100-200 contexts were gathered, and for each context, lexicographers were asked to identify which sense applied (choosing from a sense inventory in an existing dictionary), giving the set of 'correct answers'. All participants were then issued with the set

of contexts, and, within two weeks, they had to return a set of answers: the programme's estimation, for each context, of the correct word sense.

Kilgarriff & Rosenzweig (2000) show that the best performance currently achievable by automatic systems, for English, averaged over a number of words, was around 77%. Inter-human replicability for the same tasks was over 95%. The best-scoring systems all made use of a prepared corpus, with senses marked, as training data. The best results for systems not using this training data were substantially lower, around 63%. Furthermore, many of the best trainable systems had very similar scores at the top of the range suggesting that a plateau has been reached with existing techniques.

The WASP-Bench

This section outlines the system architecture and mode of operation. The workbench is implemented in perl and uses cgi-scripts and a browser for user interaction.

Grammatical Relations Database

The central resource is a collection of all grammatical relations holding between words in the corpus. The workbench is currently based on the British National Corpus[1] (BNC): 100 million words of contemporary British English, of a wide range of genres. Using finite-state techniques operating over part-of-speech tags, we process the whole corpus finding quintuples of form:

(Rel, W1, W2, Prep, Pos)

where Rel is a relation, W1 is the lemma of the word for which Rel holds, W2 is the lemma of the other open-class word involved, Prep is the preposition or particle involved and Pos is the position of W1 in the corpus. Relations may have null values for W2 and Prep. The database contains 70 million quintuples.

Relation	**Example**
bare-noun	the angle of **bank**
Possessed	my **bank**
Plural	the **bank**s
Passive	was **see**n
Reflexive	**see** herself
gerund-complement	**love** eating fish
finite-complement	**know** he came
inf-complement	**decision** to eat fish
wh-complement	**know** why he came
Subject	the **bank** refused
Object	climb the **bank**
Adjectival-complement	**grow** certain
noun-modifier	**merchant bank**
Modifier	a **big bank**
and-or	**bank**s and **mound**s
Predicate	the **bank** was a **success**
Particle	**grow up**
Prep + gerund	**tired of** eating fish
PP-comp/mod	**banks of** the **river**

Table 1: Grammatical Relations

The current inventory of relations is shown in Table 1. There are nine *unary* relations (ie. with W2 and Prep null), seven *binary* relations with Prep null, two *binary* relations with W2 null and one *trinary* relation with no null elements. All inverse relations, ie. **subject-of** etc, found by taking W2 as the head word instead of W1 are explicitly represented, giving six extra binary relations (the relation **and-or** is symmetrical and so has no inverse relation) and one extra trinary relation, to give a total of twenty-six distinct relations.

These quintuples provide a flexible resource to be used as the basis of the computations of the workbench. Keeping the position numbers of examples allows us to find associations between patterns and to display examples. The database of grammatical relations certainly contains many errors, originating from POS-tagging errors in the BNC, attachment ambiguities or limitations of the pattern-matching grammar. However, as the system finds high-salience patterns, given enough data, the noise does not present undue problems.

Word Sketches

The user enters the word of interest (together with its word class) at a prompt.[2] Using the grammatical relations database, the system then composes a **Word Sketch** for the word. This is a page of data such as Table 2, which shows, for the word in question (W1), ordered lists of high-salience grammatical relations, relation-W2 pairs, and relation-W2-Prep triples.

Subject-of	No.	Sal	**Object-of**	No.	Sal
lend	95	21.2	burst	27	16.4
issue	60	11.8	rob	31	15.3
charge	29	9.5	overflow	7	10.2
operate	45	8.9	line	13	8.4
Modifies			**PP**		
holiday	404	32.6	of England	988	37.5
account	503	32.0	of Scotland	242	26.9
loan	108	27.5	of river	111	22.1
lending	68	26.1	of Thames	41	20.1
Modifier			**Inv-PP**		
central	755	25.5	governor of	108	26.2
Swiss	87	18.7	balance at	25	20.2
commercial	231	18.6	borrow from	42	19.1
grassy	42	18.5	account with	30	18.4
n-mod			**and-or**		
merchant	213	29.4	society	287	24.6
clearing	127	27.0	bank	107	17.7
river	217	25.4	institution	82	16.0

[1] http://info.ox.ac.uk/bnc

[2] At present word classes covered are noun, verb and adjective.

Subject-of	No.	Sal	**Object-of**	No.	Sal
creditor	52	22.8	Lloyds	11	14.1

Table 2: Extract of Word Sketch for *bank* (noun)

The number of patterns shown is set by the user, but will typically be over 200. These are listed for each relation in order of salience, with the count of corpus instances. The instances can be instantly retrieved and shown in a concordance window. Producing a word sketch for a medium-to-high frequency word takes around ten seconds.[3]

Calculating Salience

Salience is estimated as the product of Mutual Information *I* (Church & Hanks, 1989) and log frequency. *I* for a (*W1,Rel,W2)* triple[4] is calculated as

$$I(W1,Rel,W2) = \log \frac{|*,Rel,*| \times |W1,Rel,W2|}{|W1,Rel,*| \times |*,Rel,W2|}$$

The notation here is adopted from Lin (1998). |W1, Rel, W2| denotes the frequency count of the triple (W1, Rel, W2)[5] in the grammatical relations database. Where W1, Rel or W2 is the wild card (*), the frequency is of all the dependency triples that match the remainder of the pattern.

Our experience of working lexicographers' use of Mutual Information or log-likelihood lists shows that, for lexicographic purposes, these over-emphasise low frequency items, and that multiplying by log frequency is an appropriate adjustment.

Matching patterns with target words

The next task is to enter a preliminary list of possible target words (or for the monolingual lexicographer, arbitrary mnemonics to represent the different possible senses). So for the word *bank* and a taget language of Russian, we might propose the target translations as **bank** (financial institution), **bereg** (river bank), **gryada** (bank of clouds) etc.

As Table 2 shows, and in keeping with ``one sense per collocation'' (Yarowsky, 1993), in most cases high-salience patterns or **clues** indicate just one of the word's senses. The user then has the task of associating, by selecting from a pop-up menu, the required target for unambiguous clues. Reference can be made at any time to the actual corpus instances, which demonstrate the contexts in which the triple occurs.[6]

[3] A set of pre-compiled word sketches can be seen at http://www.itri.bton.ac.uk/~adam.kilgarriff/wordsketches.html

[4] Grammatical-relation, preposition pairs are currently treated as atomic relations for purposes of calculating MI.

[5] Strictly, the quintuple (Rel-part-1,W1,W2,Rel-part-2, ANY).

[6] There will often be patterns that occur with more than one sense of the head word. Such is the case in Table 2 for the pattern (**subject-of**, *collapse)*, since both financial institutions and river banks collapse, albeit in different ways. The user should therefore refrain from giving a sense to this pattern: it is characteristic of *bank*, but not of just one of its senses.

Unary relations (such as **passive, finite-complement** etc) are also displayed in the word sketch and may be associated with a particular target but are more commonly used as additional conditions, positive or negative, on a target. When sufficient patterns have been marked with targets, the pattern-target pairs are submitted to the next stage: automatic disambiguation.

The Disambiguation Algorithm

The workbench currently uses Yarowsky's decision list approach to WSD (Yarowsky:, 1995). This is a bootstrapping algorithm that, given some initial seeding, iteratively divides the corpus examples into the different senses. Yarowsky notes that the most effective initial seeding option he considered was labelling salient corpus collocates with different senses. The user's first interaction with the workbench is just that.

At the user-input stage, only clues involving grammatical relations are used. At the WSD algorithm stage, some “bag-of-words” and *n*-gram clues are also considered. Any content word (lemmatised) occurring within a *k*-word window of the nodeword is a bag-of-words clue. *N*-gram clues capture local context which may not be covered by any grammatical relation. The *n*-gram clues are all bigrams and trigrams including the nodeword. A merit of the decision list approach is that probabilities are not combined, so the non-independencies of clues is not a problem.

Target Profiles

The output of the algorithm is a decision list: an ordered list of patterns each pointing to a particular target translation. These patterns will include (Rel, W2) pairs (as in the original word sketch), bag-of-words words (b-o-w), and *n*-grams. The components of the decision list which assign to a particular target can be displayed as “target profiles"', in a manner comparable to the original word sketch. The user can now review them. They will contain new clues, not originally seen in the word sketch and may point to new senses or usages needing addition to the lexical entry.

This process can be repeated until all the examples are disambiguated. The final decision list can now be used to give a target translation for the word as it appears in any context. We simply take the target given by the most highly-ranked applicable clue.

Evaluating the WASP-Bench

Lexicographic evaluation

Although the WASP-Bench is yet to be evaluated from the MT perspective, the Word Sketches have been used for monolingual lexicography. For the last two years, a set of 6000 word sketches has been used in a large dictionary project, with a team of thirty professional lexicographers using them every day, for every medium-to-high frequency noun, verb and adjective of English. The feedback we have received is that they are hugely useful, and transform the way the lexicographer uses the corpus. They radically reduce the amount of

time the lexicographers need to spend reading individual instances, and give the dictionary improved claims to completeness, as common patterns are far less likely to be missed.

WSD Evaluation

The SENSEVAL dataset comprised sets of 100-400 corpus instances for each of 41 words. Systems are required to decide on the sense of the word according to the divisions in the independently prepared HECTOR dictionary (Atkins, 1993). For most of the tasks, training data was available with senses marked for the words in question. Systems that made use of this training data (*supervised*) markedly outperformed those that did not (*unsupervised).*

We tested the WASP-Bench on the (alphabetically) first 28 of the words, recreating as closely as possible the envisaged interaction with a user-lexicographer. Referring to the dictionary entry, the user (one or other of the authors) assigned senses to clues in the word sketch, which were then submitted as seeds to the disambiguation algorithm. No further interaction was allowed. The BNC, which was the only linguistic resource used in the user-interaction phase, was disjoint from the test set. The algorithm was run only once and there was no possibility of amending the system according to performance on the test set.

The marking of senses took anywhere from 3 to 45 minutes, depending upon the subtleties of the sense divisions. It should be noted that evaluation against a predetermined sense inventory is not ideal for a flexible system such as the WASP-Bench. The system fared best where the sense distinctions were clear and could be assigned to patterns with a high degree of confidence.

The results show that overall performance for the workstation was within 1% of the best supervised system over the same subset of the data. Furthermore, the WASP-Bench achieved a better score than any other participating system in five of the twenty-eight tasks. This indicates that performance in line with the best supervised systems is possible without training data, but instead with a relatively brief interaction with the workbench.

Conclusion and further work

The original contribution of the workbench lies in bringing together corpus lexicography and WSD algorithms. The WSD algorithm is shown to be on a par with the very best systems, without requiring training data.

The workbench addresses the usual situation for MT companies, in which the starting point, particularly for a new language-pair, is typically a lexical resource needing a large amount of manual input before it supports acceptable-quality translation, and the task thereafter is one of continual lexicon maintenance and improvement. The workbench is designed to work with an existing resource that needs improvement--and MT companies are familiar with the fact that high-quality translation takes many lexicographer-years.

We do of course wish to close the gap between the 95% human-human agreement on WSD, and the 70-80% best current system performance. We shall be adding a data-driven thesaurus (Lin, 1998) to the workbench, which will firstly permit the specification of clues as relating to thesaural categories as well as individual words. Other plans include developing the potential for using web data, with pages being downloaded and fed directly into the workbench. This strategy would extend the potential of the workbench beyond source languages where large corpora are readily available.

References

Atkins, S. (1993). Tools for computer-aided corpus lexicography: the Hector project. *Acta Linguistica Hungarica*, 41, 5-72.

Church, K. & Hanks, P. (1989). Word association norms, mutual information and lexicography. In *ACL Proceedings, 27th Annual Meeting*, 76-83, Vancouver.

Grefenstette, G. & Nioche, J. (2000). Estimation of English and non-English Language Use on the WWW. In *Proc. RIAO (Recherche d'Informations Assistee par Ordinateur),* Paris.

Ide, N & Veronis, J. (1998). Introduction to the Special Issue on Word Sense Disambiguation: The state of the art. *Computational Linguistics, 24, 1:* 1-40.

Kilgarriff, A. (1997). What is word sense disambiguation good for? In *Proc. Natural Language Processing in the Pacific Rim (NLPRS '97)*, Phuket, Thailand.

Kilgarriff, A. & Rosenzweig, J. (2000). Framework and results for English SENSEVAL. *Computers and the Humanities*, 34(1-2): 15-48.

Kilgarriff, A. & Palmer, M. (2000). Introduction, Special Issue on SENSEVAL: Evaluating Word Sense Disambiguation Programs. *Computers and the Humanities*, 34(1-2):1-13.

Lin, D.K. (1998). Automatic retrieval and clustering of similar words. In COLING-ACL, 768-774, Montreal.

Yarowsky, D. (1993). One sense per collocation. In *Proc. ARPA Human Language Technology Workshop*, Princeton.

Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. In *ACL 95*, 189-196, MIT.