Detecting Multi-Word Expressions improves Word Sense Disambiguation

Mark Alan Finlayson & Nidhi Kulkarni
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA, 02139, USA
{markaf,nidhik}@mit.edu

Abstract

Multi-Word Expressions (MWEs) are prevalent in text and are also, on average, less polysemous than mono-words. This suggests that accurate MWE detection should lead to a non-trivial improvement in Word Sense Disambiguation (WSD). We show that a straightforward MWE detection strategy, due to Arranz *et al.* (2005), can increase a WSD algorithm's baseline f-measure by 5 percentage points. Our measurements are consistent with Arranz's, and our study goes further by using a portion of the Semcor corpus containing 12,449 MWEs - over 30 times more than the approximately 400 used by Arranz. We also show that perfect MWE detection over Semcor only nets a total 6 percentage point increase in WSD f-measure; therefore there is little room for improvement over the results presented here. We provide our MWE detection algorithms, along with a general detection framework, in a free, open-source Java library called jMWE.

Multi-word expressions (MWEs) are prevalent in text. This is important for the classic task of Word Sense Disambiguation (WSD) (Agirre and Edmonds, 2007), in which an algorithm attempts to assign to each word in a text the appropriate entry from a sense inventory. A WSD algorithm that cannot correctly detect the MWEs that are listed in its sense inventory will not only miss those sense assignments, it will also spuriously assign senses to MWE constituents that themselves have sense entries, dealing a double-blow to WSD performance.

Beyond this penalty, MWEs listed in a sense inventory also present an opportunity to WSD algorithms - they are, on average, less polysemous than mono-words. In Wordnet 1.6, multi-words have an average polysemy of 1.07, versus 1.53 for mono-words. As a concrete example, consider sentence *She broke the world record*. In Wordnet 1.6 the lemma *world* has nine different senses and *record* has fourteen, while the MWE *world record* has only one. If a WSD algorithm correctly detects MWEs, it can dramatically reduce the number of possible senses for such sentences.

Measure	**Us**	**Arranz**
Number of MWEs	12,449	382
Fraction of MWEs	7.4%	9.4%
WSD impr. (Best v. Baseline)	0.016_{F_1}	0.012_{F_1}
WSD impr. (Baseline v. None)	0.033_{F_1}	-
WSD impr. (Best v. None)	$\mathbf{0.050}_{F_1}$	-
WSD impr. (Perfect v. None)	$\mathbf{0.061}_{F_1}$	-

Table 1: Improvement of WSD f-measures over an MWE-unaware WSD strategy for various MWE detection strategies. *Baseline*, *Best*, and *Perfect* refer to the MWE detection strategy used in the WSD preprocess.

With this in mind, we expected that accurate MWE detection will lead to a small yet non-trivial improvement in WSD performance, and this is indeed the case. Table 1 summarizes our results. In particular, a relatively straightforward MWE detection strategy, here called the 'best' strategy and due to Arranz *et al.* (2005), yielded a 5 percentage point improvement[1] in WSD f-measure. We also measured an improvement similar to that of Arranz when

[1] For example, if the WSD algorithm has an f-measure of

Proceedings of the Workshop on Multiword Expressions: from Parsing and Generation to the Real World (MWE 2011), pages 20–24,
Portland, Oregon, USA, 23 June 2011. ©2011 Association for Computational Linguistics

moving from a Baseline MWE detection strategy to the Best strategy, namely, 1.6 percentage points to their 1.2.

We performed our measurements over the *brown1* and *brown2* concordances[2] of the Semcor corpus (Fellbaum, 1998), which together contain 12,449 MWEs, over 30 times as many as the approximately 400 contained in the portion of the XWN corpus used by Arranz. We also measured the improvement for WSD f-measure for Baseline and Perfect MWE detection strategies. These strategies improved WSD f-measure by 3.3 and 6.1 percentage points, respectively, showing that the relatively straightforward Best MWE detection strategy, at 5.0 percentage points, leaves little room for improvement.

1 MWE Detection Algorithms by Arranz

Arranz *et al.* describe their TALP Word Sense Disambiguation system in (Castillo et al., 2004) and (Arranz et al., 2005). The details of the WSD procedure are not critical here; what is important is that their preprocessing system attempted to detect MWEs that could later be disambiguated by the WSD algorithm. This preprocessing occurred as a pipeline that tokenized the text, assigned a part-of-speech tag, and finally determined a lemma for each stemmable word. This information was then passed to a MWE candidate identifier[3] whose output was then filtered by an MWE selector. The resulting list of MWEs, along with all remaining tokens, were then passed into the WSD algorithm for disambiguation.

The MWE identifier-selector pair determined what combinations of tokens were marked as MWEs. It considered only continuous (i.e., unbroken) sequences of tokens whose order matched the order of the constituents of the associated MWE entry in Wordnet. Because of morphological variation, not all sequences of tokens are in base form; the main function of the candidate identifier, therefore, was to determine what morphological variation was allowed for a particular MWE entry. They identified and tested four different strategies:

1. **None** - no morphological variation allowed, all MWEs must be in base form
2. **Pattern** - allows morphological variation according to a set of pre-defined patterns
3. **Form** - a morphological variant is allowed if it is observed in Semcor
4. **All** - all morphological variants allowed

The identification procedure produced a list of candidate MWEs. These MWEs were then filtered by the MWE selection process, which used one of two strategies:

1. **Longest Match, Left-to-Right** - starting from the left to right, selects the longest multi-word expression found
2. **Semcor** - selects the multi-word expression whose tokens have the maximum probability of participating in an MWE, according to measurements over Semcor

Arranz identified the *None/Longest-Match-Left-to-Right* strategy as the Baseline, noting that this was the most common strategy for MWE-aware WSD algorithms. For this strategy the only MWE candidates allowed were those already in base form (*None*), followed by resolution of conflicts by selecting the MWEs that started farthest to the left, choosing the longest in case of ties (*Longest-Match-Left-to-Right*);

Arranz's Best strategy was *Pattern/Semcor*, namely, allowing candidate MWEs to vary morphologically according to a pre-defined set of syntactic patterns (*Pattern*), followed by selecting the most likely MWE based on examination of token frequencies in the Semcor corpus (*Semcor*). They ran their detection strategies over a subset of the sense-disambiguated glosses of the eXtended WordNet (XWN) corpus (Moldovan and Novischi, 2004). They selected all glosses whose sense-disambiguated words were all marked as 'gold' quality, namely, reviewed by a human annotator. Over this set of words, their WSD system achieved 0.617_{F_1} (0.622_p/0.612_r) when using the Baseline MWE detection strategy, and 0.629_{F_1} (0.638_p/0.620_r) when using the Best strategy.

0.6, then a 5 percentage point increase yields an f-measure of 0.65.

[2] The third concordance, *brownv*, only has verbs marked, so we did not test on it.

[3] Arranz calls the candidate identification stage the MWE *detector*; we have renamed it because we take 'detection' to be the end-to-end process of marking MWEs.

2 Extension of Results

We implemented both the Baseline and Best MWE-detection strategies, and used them as preprocessors for a simple WSD algorithm, namely, the Most-Frequent Sense algorithm. This algorithm simply chooses, for each identified base form, the most frequent sense in the sense inventory. We chose this strategy, instead of re-implementing Arranz's strategy, for two reasons. First, our purpose was to study the improvement MWE-detection provides to WSD in general, not to a specific WSD algorithm. We wished to show that, to the first order, MWE detection improves WSD irrespective of the WSD algorithm chosen. Using a different algorithm than Arranz's supports this claim. Second, for those wishing to further this work, or build upon it, the Most-Frequent-Sense strategy is easily implemented.

We used JSemcor (Finlayson, 2008a) to interface with the Semcor data files. We used Wordnet version 1.6 with the original version of Semcor[4]. Each token in each sentence in the *brown1* and *brown2* concordances of Semcor was assigned a part of speech tag calculated using the Stanford Java NLP library (Toutanova et al., 2003), as well as a set of lemmas calculated using the MIT Java Wordnet Interface (Finlayson, 2008b). This data was the input to each MWE detection strategy.

There was one major difference between our detector implementations and Arranz, stemming from a major difference between XWN and Semcor: Semcor contains a large number of proper nouns, whereas XWN glosses contain almost none. Therefore our detector implementations included a simple proper noun MWE detector, which marked all unbroken runs of tokens tagged as proper nouns as a proper noun MWE. This proper noun detector was run first, before the Baseline and Best detectors, and the proper noun MWEs detected took precedence over the MWEs detected in later stages.

Baseline MWE Detection This MWE detection strategy was called *None/Longest-Match-Left-to-Right* by Arranz; we implemented it in four stages. First, we detected proper nouns, as described. Second, for each sentence, the strategy used the part of speech tags and lemmas to identify all possible consecutive MWEs, using a list extracted from WordNet 1.6 and Semcor 1.6. The only restriction was that at least one token identified as part of the MWE must share the basic part of speech (e.g., noun, verb, adjective, or adverb) with the part of speech of the MWE. As noted, tokens that were identified as being part of a proper noun MWE were not included in this stage. In the third stage, we removed all non-proper-noun MWEs that were inflected–this corresponds to Arranz's *None* stage. In our final stage, any conflicts were resolved by choosing the MWE with the leftmost token. For two conflicting MWEs that started at the same token, the longest MWE was chosen. This corresponds to Arranz's *Longest-Match-Left-to-Right* selection.

Best MWE Detection This MWE detection strategy was called *Pattern/Semcor* by Arranz, and we also implemented this strategy in four stages. The first and second stages were the same as the Baseline strategy, namely, detection of proper nouns followed by identification of continuous MWEs. The third stage kept only MWEs whose morphological inflection matched one of the inflection rules described by Arranz (*Pattern*). The final stage resolved any conflicts by choosing the MWE whose constituent tokens occur most frequently in Semcor as an MWE rather than a sequence of monowords (Arranz's *Semcor* selection).

Word Sense Disambiguation No special technique was required to chain the Most-Frequent Sense WSD algorithm with the MWE detection strategies. We measured the performance of the WSD algorithm using no MWE detection, the Baseline detection, the Best detection, and Perfect detection[5]. These results are shown in Table 2.

Our improvement from Baseline to Best was approximately the same as Arranz's: 1.7 percentage points to their 1.2. We attribute the difference to the much worse performance of our Baseline detection algorithm: our Baseline MWE detection f-measure was 0.552, compared their 0.740. The reason for this

[4] The latest version of Wordnet is 3.0, but Semcor has not been manually updated for Wordnet versions later than 1.6. Automatically updated versions of Semcor are available, but they contain numerous errors resulting from deleted sense entries, and the sense assignments and multi-word identifications have not been adjusted to take into account new entries. Therefore we decided to use versions 1.6 for both Wordnet and Semcor.

[5] Perfect detection merely returned the MWEs identified in Semcor

Measure	**Arranz *et al.* (2005)**	**Finlayson & Kulkarni**
Corpus	eXtended WordNet (XWN) 2.0	Semcor 1.6 (`brown1` & `brown2`)
Number of Tokens (non-punctuation)	8,493	376,670
Number of Open-Class Tokens	5,133	196,852
Number of Open-Class Monowords	4,332	168,808
Number of Open-Class MWEs	382	12,449
Number of Tokens in Open-Class MWEs	801	28,044
Number of Open-Class Words (mono & multi)	4,714	181,257
Fraction MWEs	9.4%	7.4%
MWE Detection, Baseline	0.740_{F_1} $(0.765_p/0.715_r)$	0.552_{F_1} $(0.452_p/0.708_r)$
MWE Detection, Best	0.811_{F_1} $(0.806_p/0.816_r)$	0.856_{F_1} $(0.874_p/0.838_r)$
WSD, MWE-unaware	-	0.579_{F_1} $(0.572_p/0.585_r)$
WSD, Baseline MWE Detection	0.617_{F_1} $(0.622_p/0.612_r)$	0.612_{F_1} $(0.614_p/0.611_r)$
WSD, Best MWE Detection	0.629_{F_1} $(0.638_p/0.620_r)$	0.629_{F_1} $(0.630_p/0.628_r)$
WSD, Perfect MWE Detection	-	0.640_{F_1} $(0.642_p/0.638_r)$
WSD Improvement, Baseline vs. Best	0.012_{F_1} $(0.016_p/0.008_r)$	0.016_{F_1} $(0.016_p/0.017_r)$
WSD Improvement, Baseline vs. None	-	$\mathbf{0.033}_{F_1}$ $\mathbf{(0.042}_p\mathbf{/0.025}_r\mathbf{)}$
WSD Improvement, Best vs. None	-	$\mathbf{0.050}_{F_1}$ $\mathbf{(0.058}_p\mathbf{/0.043}_r\mathbf{)}$
WSD Improvement, Perfect vs. None	-	$\mathbf{0.061}_{F_1}$ $\mathbf{(0.070}_p\mathbf{/0.053}_r\mathbf{)}$

Table 2: All the relevant numbers for the study. For purposes of comparison we recalculated the token counts for the gold-annotated portion of the XWN corpus, and found discrepancies with Arranz's reported values. They reported 1300 fully-gold-annotated glosses containing 397 MWEs; we found 1307 glosses containing 382 MWEs. The table contains our token counts, but Arranz's actual MWE detection and WSD f-measures, precisions, and recalls.

striking difference in Baseline performance seems to be that, in the XWN glosses, a much higher fraction of the MWEs are already in base form (e.g., nouns in glosses are preferentially expressed as singular).

To encourage other researchers to build upon our results, we provide our implementation of these two MWE detection strategies, along with a general MWE detection framework and numerous other MWE detectors, in the form of a free, open-source Java library called jMWE (Finlayson and Kulkarni, 2011a). Furthermore, to allow independent verification of our results, we have placed all the source code and data required to run these experiments in an online repository (Finlayson and Kulkarni, 2011b).

3 Contributions

We have shown that accurately detecting multiword expressions allows a non-trivial improvement in word sense disambiguation. Our Baseline, Best, and Perfect MWE detection strategies show a 3.3, 5.1, and 6.1 percentage point improvement in WSD f-measure. Our Baseline-to-Best improvement is comparable with that measured by Arranz, the difference being due to more prevalent base-form MWEs between XWN glosses and Semcor. The very small improvement of the Perfect strategy over the Best shows that, at least for Wordnet over texts with an MWE distribution similar to Semcor, there is little to be gained even from a highly-sophisticated MWE detector. We have provided these two MWE detection algorithms in a free, open-source Java library called jMWE.

Acknowledgments

This work was supported in part by the Air Force Office of Scientific Research under grant number A9550-05-1-0321, and the Defense Advanced Research Projects Agency under contract number FA8750-10-1-0076. Thanks to Michael Fay for helpful comments.

References

Eneko Agirre and Philip Edmonds. 2007. *Word Sense Disambiguation*. Text, Speech, and Language Technology. Springer-Verlag, Dordrecht, The Netherlands.

Victoria Arranz, Jordi Atserias, and Mauro Castillo. 2005. Multiwords and word sense disambiguation. In Alexander Gelbukh, editor, *Proceedings of the Sixth International Conference on Intelligent Text Processing and Computational Linguistics (CICLING)*, volume 3406 of Lecture Notes in Computer Science (LNCS), pages 250–262, Mexico City, Mexico. Springer-Verlag.

Mauro Castillo, Francis Real, Jordi Asterias, and German Rigau. 2004. The TALP systems for disambiguating WordNet glosses. In Rada Mihalcea and Phil Edmonds, editors, *Proceedings of Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 93–96. Association for Computational Linguistics.

Christiane Fellbaum. 1998. *Wordnet: An Electronic Lexical Database*. MIT Press, Cambridge, MA.

Mark Alan Finlayson and Nidhi Kulkarni. 2011a. jMWE, version 1.0.0.
http://projects.csail.mit.edu/jmwe
http://hdl.handle.net/1721.1/62793.

Mark Alan Finlayson and Nidhi Kulkarni. 2011b. Source code and data for MWE'2011 papers.
http://hdl.handle.net/1721.1/62792.

Mark Alan Finlayson. 2008a. JSemcor, version 1.0.0.
http://projects.csail.mit.edu/jsemcor.

Mark Alan Finlayson. 2008b. JWI: The MIT Java Wordnet Interface, version 2.1.5.
http://projects.csail.mit.edu/jwi.

Dan Moldovan and Adrian Novischi. 2004. Word sense disambiguation of WordNet glosses. *Computer Speech and Language*, 18:301–317.

Kristina Toutanova, Daniel Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. pages 252–259. Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL).