The Effects of Lexical Resource Quality on Preference Violation Detection

Jesse Dunietz
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA, 15213, USA
jdunietz@cs.cmu.edu

Lori Levin and **Jaime Carbonell**
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA, 15213, USA
{lsl,jgc}@cs.cmu.edu

Abstract

Lexical resources such as WordNet and VerbNet are widely used in a multitude of NLP tasks, as are annotated corpora such as treebanks. Often, the resources are used as-is, without question or examination. This practice risks missing significant performance gains and even entire techniques.

This paper addresses the importance of resource quality through the lens of a challenging NLP task: detecting selectional preference violations. We present DAVID, a simple, lexical resource-based preference violation detector. With as-is lexical resources, DAVID achieves an F_1-measure of just 28.27%. When the resource entries and parser outputs for a small sample are corrected, however, the F_1-measure on that sample jumps from 40% to 61.54%, and performance on other examples rises, suggesting that the algorithm becomes practical given refined resources. More broadly, this paper shows that resource quality matters tremendously, sometimes even more than algorithmic improvements.

1 Introduction

A variety of NLP tasks have been addressed using *selectional preferences* or *restrictions*, including word sense disambiguation (see Navigli (2009)), semantic parsing (e.g., Shi and Mihalcea (2005)), and metaphor processing (see Shutova (2010)). These semantic problems are quite challenging; metaphor analysis, for instance, has long been recognized as requiring considerable semantic knowledge (Wilks, 1978; Carbonell, 1980). The advent of extensive lexical resources, annotated corpora, and a spectrum of NLP tools presents an opportunity to revisit such challenges from the perspective of selectional preference violations. Detecting these violations, however, constitutes a severe stress-test for resources designed for other tasks. As such, it can highlight shortcomings and allow quantifying the potential benefits of improving resources such as WordNet (Fellbaum, 1998) and VerbNet (Schuler, 2005).

In this paper, we present DAVID (Detector of Arguments of Verbs with Incompatible Denotations), a resource-based system for detecting preference violations. DAVID is one component of METAL (Metaphor Extraction via Targeted Analysis of Language), a new system for identifying, interpreting, and cataloguing metaphors. One purpose of DAVID was to explore how far lexical resource-based techniques can take us. Though our initial results suggested that the answer is "not very," further analysis revealed that the problem lies less in the technique than in the state of existing resources and tools.

Often, it is assumed that the frontier of performance on NLP tasks is shaped entirely by algorithms. Manning (2011) showed that this may not hold for POS tagging – that further improvements may require resource cleanup. In the same spirit, we argue that for some semantic tasks, exemplified by preference violation detection, resource quality may be at least as essential as algorithmic enhancements.

2 The Preference Violation Detection Task

DAVID builds on the insight of Wilks (1978) that the strongest indicator of metaphoricity is the violation of selectional preferences. For example, only plants can literally be pruned. If *laws* is the object of *pruned*, the verb is likely metaphorical. Flagging such semantic mismatches between verbs and arguments is the task of preference violation detection.

Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pages 765–770,
Sofia, Bulgaria, August 4-9 2013. ©2013 Association for Computational Linguistics

We base our definition of preferences on the Pragglejaz guidelines (Pragglejaz Group, 2007) for identifying the most basic sense of a word as the most concrete, embodied, or precise one. Similarly, we define selectional preferences as the semantic constraints imposed by a verb's most basic sense. Dictionaries may list figurative senses of *prune*, but we take the basic sense to be cutting plant growth.

Several types of verbs were excluded from the task because they have very lax preferences. These include verbs of becoming or seeming (e.g., *transform*, *appear*), light verbs, auxiliaries, and aspectual verbs. For the sake of simplifying implementation, phrasal verbs were also ignored.

3 Algorithm Design

To identify violations, DAVID employs a simple algorithm based on several existing tools and resources: SENNA (Collobert et al., 2011), a semantic role labeling (SRL) system; VerbNet, a computational verb lexicon; SemLink (Loper et al., 2007), which includes mappings between PropBank (Palmer et al., 2005) and VerbNet; and WordNet. As one metaphor detection component of METAL's several, DAVID is designed to favor precision over recall. The algorithm is as follows:

1. Run the Stanford CoreNLP POS tagger (Toutanova et al., 2003) and the TurboParser dependency parser (Martins et al., 2011).
2. Run SENNA to identify the semantic arguments of each verb in the sentence using the PropBank argument annotation scheme (`Arg0`, `Arg1`, etc.). See Table 1 for example output.
3. For each verb V, find all VerbNet entries for V. Using SemLink, map each PropBank argument name to the corresponding VerbNet thematic roles in these entries (Agent, Patient, etc.). For example, the VerbNet class for *prune* is `carve-21.2-2`. SemLink maps `Arg0` to the Agent of `carve-21.2-2` and `Arg1` to the Patient.
4. Retrieve from VerbNet the selectional restrictions of each thematic role. In our running example, VerbNet specifies `+int_control` and `+concrete` for the Agent and Patient of `carve-21.2-2`, respectively.
5. If the head of any argument cannot be interpreted to meet V's preferences, flag V as a violation.

"The politician pruned laws regulating plastic bags, and created new fees for inspecting dairy farms."

Verb	Arg0	Arg1
pruned	The politician	laws . . . bags
regulating	laws	plastic bags
created	The politician	new fees
inspecting	- -	dairy farms

Table 1: SENNA's SRL output for the example sentence above. Though this example demonstrates only two arguments, SENNA is capable of labeling up to six.

Restriction	WordNet Synsets
animate	`animate_being.n.01`
	`people.n.01`
	`person.n.01`
concrete	`physical_object.n.01`
	`matter.n.01`
	`substance.n.01`
organization	`social_group.n.01`
	`district.n.01`

Table 2: DAVID's mappings between some common VerbNet restriction types and WordNet synsets.

Each VerbNet restriction is interpreted as mandating or forbidding a set of WordNet hypernyms, defined by a custom mapping (see Table 2). For example, VerbNet requires both the Patient of a verb in `carve-21.2-2` and the Theme of a verb in `wipe_manner-10.4.1-1` to be concrete. By empirical inspection, concrete nouns are hyponyms of the WordNet synsets `physical_object.n.01`, `matter.n.03`, or `substance.n.04`. *Laws* (the Patient of *prune*) is a hyponym of none of these, so *prune* would be flagged as a violation.

4 Corpus Annotation

To evaluate our system, we assembled a corpus of 715 sentences from the METAL project's corpus of sentences with and without metaphors. The corpus was annotated by two annotators following an annotation manual. Each verb was marked for whether its arguments violated the selectional preferences of the most basic, literal meaning of the verb. The annotators resolved conflicts by dis-

Error source	Frequency
Bad/missing VN entries	4.5 (14.1%)
Bad/missing VN restrictions	6 (18.8%)
Bad/missing SL mappings	2 (6.3%)
Parsing/head-finding errors	3.5 (10.9%)
SRL errors	8.5 (26.6%)
VN restriction system too weak	4 (12.5%)
Confounding WordNet senses	3.5 (10.9%)
Endemic errors:	7.5 (23.4%)
Resource errors:	12.5 (39.1%)
Tool errors:	12 (37.5%)
Total:	32 (100%)

Table 3: Sources of error in 90 randomly selected sentences. For errors that were due to a combination of sources, 1/2 point was awarded to each source. (VN stands for VerbNet and SL for SemLink.)

cussing until consensus.

5 Initial Results

As the first row of Table 4 shows, our initial evaluation left little hope for the technique. With such low precision and F_1, it seemed a lexical resource-based preference violation detector was out. When we analyzed the errors in 90 randomly selected sentences, however, we found that most were not due to systemic problems with the approach; rather, they stemmed from SRL and parsing errors and missing or incorrect resource entries (see Table 3). Armed with this information, we decided to explore how viable our algorithm would be absent these problems.

6 Refining The Data

To evaluate the effects of correcting DAVID's inputs, we manually corrected the tool outputs and resource entries that affected the aforementioned 90 sentences. SRL output was corrected for every sentence, while SemLink and VerbNet entries were corrected only for each verb that produced an error.

6.1 Corrections to Tool Output (Parser/SRL)

Guided by the PropBank database and annotation guidelines, we corrected all errors in core role assignments from SENNA. These corrections included relabeling arguments, adding missed arguments, fixing argument spans, and deleting annotations for non-verbs. The only parser-related error we corrected was a mislabeled noun.

6.2 Correcting Corrupted Data in VerbNet

The VerbNet download is missing several subclasses that are referred to by SemLink or that have been updated on the VerbNet website. Some roles also have not been updated to the latest version, and some subclasses are listed with incorrect IDs. These problems, which caused SemLink mappings to fail, were corrected before reviewing errors from the corpus.

Six subclasses needed to be fixed, all of which were easily detected by a simple script that did not depend on the 90-sentence subcorpus. We therefore expect that few further changes of this type would be needed for a more complete resource refinement effort.

6.3 Corpus-Based Updates to SemLink

Our modifications to SemLink's mappings included adding missing verbs, adding missing roles to mappings, and correcting mappings to more appropriate classes or roles. We also added null mappings in cases where a PropBank argument had no corresponding role in VerbNet. This makes the system's strategy for ruling out mappings more reliable.

No corrections were made purely based on the sample. Any time a verb's mappings were edited, VerbNet was scoured for plausible mappings for every verb sense in PropBank, and any nonsensical mappings were deleted. For example, when the phrase *go dormant* caused an error, we inspected the mappings for *go*. Arguments of all but 2 of the 7 available mappings were edited, either to add missing arguments or to correct nonsensical ones. These changes actually had a net negative impact on test set performance because the bad mappings had masked parsing and selectional preference problems.

Based on the 90-sentence subcorpus, we modified 20 of the existing verb entries in SemLink. These changes included correcting 8 role mappings, adding 13 missing role mappings to existing senses, deleting 2 incorrect senses, adding 11 verb senses, correcting 2 senses, deleting 1 superfluous role mapping, and adding 46 null role mappings. (Note that although null mappings represented the largest set of changes, they also had the least impact on system behavior.) One entirely new verb was added, as well.

6.4 Corpus-Based Updates to VerbNet

Nineteen VerbNet classes were modified, and one class had to be added. The modifications generally involved adding, correcting, or deleting selectional restrictions, often by introducing or rearranging subclasses. Other changes amounted to fixing clerical errors, such as incorrect role names or restrictions that had been ANDed instead of ORed.

An especially difficult problem was an inconsistency in the semantics of VerbNet's subclass system. In some cases, the restrictions specified on a verb in a subclass did not apply to subcategorization frames inherited from a superclass, but in other cases the restrictions clearly applied to all frames. The conflict was resolved by duplicating subclassed verbs in the top-level class whenever different selectional restrictions were needed for the two sets of frames.

As with SemLink, samples determined only *which* classes were modified, not what modifications were made. Any non-obvious changes to selectional restrictions were verified by examining dozens of verb instances from SketchEngine's (Kilgarriff et al., 2004) corpus. For example, the Agent of *seek* was restricted to `+animate`, but the corpus confirmed that organizations are commonly described non-metaphorically as seeking, so the restriction was updated to `+animate | +organization`.

7 Results After Resource Refinement

After making corrections for each set of 10 sentences, we incrementally recomputed F_1 and precision, both on the subcorpus corrected so far and on a test set of all 625 sentences that were never corrected. (The manual nature of the correction effort made testing k-fold subsets impractical.) The results for 30-sentence increments are shown in Table 4.

The most striking feature of these figures is how much performance improves on corrected sentences: for the full 90 sentences, F_1 rose from 30.43% to 61.54%, and precision rose even more dramatically from 31.82% to 80.00%. Interestingly, resource corrections alone generally made a larger difference than tool corrections alone, suggesting that resources may be the dominant factor in resource-intensive tasks such as this one. Even more compellingly, the improvement from correcting both the tools and the resources was nearly double the sum of the improvements from each alone: tool and resource improvements interact synergistically.

The effects on the test corpus are harder to interpret. Due to a combination of SRL problems and the small number of sentences corrected, the scores on the test set improved little with resource correction; in fact, they even dipped slightly between the 30- and 60-sentence increments. Nonetheless, we contend that our results testify to the generality of our corrections: after each iteration, every altered result was either an error fixed or an error that should have appeared before but had been masked by another. Note also that all results on the test set are without corrected tool output; presumably, these sentences would also have improved synergistically with more accurate SRL. How long corrections would continue to improve performance is a question that we did not have the resources to answer, but our results suggest that there is plenty of room to go.

Some errors, of course, are endemic to the approach and cannot be fixed either by improved resources or by better tools. For example, we consider every WordNet sense to be plausible, which produces false negatives. Additionally, the selectional restrictions specified by VerbNet are fairly loose; a more refined set of categories might capture the range of verbs' restrictions more accurately.

8 Implications for Future Refinement Efforts

Although improving resources is infamously labor-intensive, we believe that similarly refining the remainder of VerbNet and SemLink would be doable. In our study, it took about 25-35 person-hours to examine about 150 verbs and to modify 20 VerbNet classes and 25 SemLink verb entries (excluding time for SENNA corrections, fixing corrupt VerbNet data, and analysis of DAVID's errors). Extrapolating from our experience, we estimate that it would take roughly 6-8 person-weeks to systematically fix this particular set of issues with VerbNet.

Improving SemLink could be more complex, as its mappings are automatically generated from VerbNet annotations on top of the PropBank corpus. One possibility is to correct the generated mappings directly, as we did in our study, which we estimate would take about two person-months.

With the addition of some metadata from the generation process, it would then be possible to follow the corrected mappings back to annotations from which they were generated and fix those annotations. One downside of this approach is that if the mappings were ever regenerated from the annotated corpus, any mappings not encountered in the corpus would have to be added back afterwards.

Null role mappings would be particularly thorny to implement. To add a null mapping, we must know that a role definitely does not belong, and is not just incidentally missing from an example. For instance, VerbNet's `defend-85` class truly has no equivalent to `Arg2` in PropBank's `defend.01`, but `Arg0` or `Arg1` may be missing for other reasons (e.g., in a passive). It may be best to simply omit null mappings, as is currently done. Alternatively, full parses from the Penn Treebank, on which PropBank is based, might allow distinguishing phenomena such as passives where arguments are predictably omitted.

The maintainers of VerbNet and PropBank are aware of many of the issues we have raised, and we have been in contact with them about possible approaches to fixing them. They are particularly aware of the inconsistent semantics of selectional restrictions on VerbNet subclasses, and they hope to fix this issue within a larger attempt at retooling VerbNet's selectional restrictions. In the meantime, we are sharing our VerbNet modifications with them for them to verify and incorporate. We are also sharing our SemLink changes so that they can, if they choose, continue manual correction efforts or trace SemLink problems back to the annotated corpus.

9 Conclusion

Our results argue for investing effort in developing and fixing resources, in addition to developing better NLP tools. Resource and tool improvements interact synergistically: better resources multiply the effect of algorithm enhancements. Gains from fixing resources may sometimes even exceed what the best possible algorithmic improvements can provide. We hope the NLP community will take up the challenge of investing in its resources to the extent that its tools demand.

Sent.	**Tools**	**Rsrcs**	**P**	$\mathbf{F_1}$
715	0	0	27.14%	28.27%
625	0	0	26.55%	27.98%
625	0	corr.	26.37%	28.15%
30	0	0	50.00%	40.00%
30	30	0	66.67%	44.44%
30	0	corr.+30	62.50%	50.00%
30	30	corr.+30	87.50%	70.00%
625	0	corr.+30	27.07%	28.82%
60	0	0	35.71%	31.25%
60	60	0	54.55%	31.38%
60	0	corr.+60	53.85%	45.16%
60	60	corr.+60	90.91%	68.97%
625	0	corr.+60	26.92%	28.74%
90	0	0	31.82%	30.43%
90	90	0	44.44%	38.10%
90	0	corr.+90	47.37%	41.86%
90	90	corr.+90	80.00%	61.54%
625	0	corr.+90	27.37%	28.99%

Table 4: Performance on preference violation detection task. Column 1 shows the sentence count. Columns 2 and 3 show how many sentences' SRL/parsing and resource errors, respectively, had been fixed ("corr." indicates corrupted files).

Acknowledgments

Thanks to Eric Nyberg for suggesting building a system like DAVID, to Spencer Onuffer for his annotation efforts, and to Davida Fromm for curating METAL's corpus of Engish sentences.

This work was supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense US Army Research Laboratory contract number W911NF-12-C-0020. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoD/ARL, or the U.S. Government.

References

Jaime G. Carbonell. 1980. Metaphor: a key to extensible semantic analysis. In *Proceedings of the 18th annual meeting on Association for Computational Linguistics*, ACL '80, pages 17–21, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ronan Collobert, Jason Weston, Léon Bottou, Michael

Karlen, Koray Kavukcuoglu, and Pavel P. Kuksa. 2011. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12:2493–2537, November.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. Bradford Books.

Adam Kilgarriff, Pavel Rychly, Pavel Smrz, and David Tugwell. 2004. The Sketch Engine. In *Proceedings of EURALEX*.

Edward Loper, Szu-ting Yi, and Martha Palmer. 2007. Combining lexical resources: Mapping between PropBank and VerbNet. In *Proceedings of the 7th International Workshop on Computational Linguistics, Tilburg, the Netherlands*.

Christopher D Manning. 2011. Part-of-speech tagging from 97% to 100%: is it time for some linguistics? In *Computational Linguistics and Intelligent Text Processing*, pages 171–189. Springer.

André F. T. Martins, Noah A. Smith, Pedro M. Q. Aguiar, and Mário A. T. Figueiredo. 2011. Dual decomposition with many overlapping components. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 238–249, Stroudsburg, PA, USA. Association for Computational Linguistics.

Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM Computing Surveys (CSUR)*, 41(2):10.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

Pragglejaz Group. 2007. MIP: A method for identifying metaphorically used words in discourse. *Metaphor and Symbol*, 22(1):1–39.

Karin K. Schuler. 2005. *VerbNet: A Broad-Coverage, Comprehensive Verb Lexicon*. Ph.D. thesis, University of Pennsylvania, Philadelphia, PA. AAI3179808.

Lei Shi and Rada Mihalcea. 2005. Putting pieces together: Combining FrameNet, VerbNet and WordNet for robust semantic parsing. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*, volume 3406 of *Lecture Notes in Computer Science*, pages 100–111. Springer Berlin Heidelberg.

Ekaterina Shutova. 2010. Models of metaphor in NLP. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 688–697, Stroudsburg, PA, USA. Association for Computational Linguistics.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 173–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

Yorick Wilks. 1978. Making preferences more active. *Artificial Intelligence*, 11:197–223.