Is Three the Optimal Context Window for Memory-Based Word Sense Disambiguation?

Rodrigo de Oliveira, Lucas Hausmann and Desislava Zhekova
University of Bremen
(rdeoliveira, lhausmann, zhekova) @uni-bremen.de

Abstract

In this work we research the effect of micro-context on a memory-based learning (MBL) system for word sense disambiguation. We report results achieved on the data set provided by the English Lexical Sample Task introduced in the Senseval 3 competition. Our study revisits the belief that the disambiguation task profits more from a wider context and indicates that in reality system performance is highest when a narrower context is considered.

Keywords

word sense disambiguation, memory-based learning, supervised learning

1 Introduction

Back in the 50's since the first efforts in computational linguistics, it has been said that more context information leads to a stronger guiding in resolving the problem of ambiguity (Weaver, 1955). Yet, there are different kinds of ambiguity (e.g. structural vs. lexical ambiguity). Word sense disambiguation (WSD), as reviewed by Ide and Véronis (1998), is targeting the problem of lexical ambiguity. In general, it aims to find the correct sense of a given word depending on the context in which it is found. According to the authors, context can also be defined in different ways: micro-context, constructed by a window of n (e.g. 1, 2, 3, etc.) number of words before and after the target word; topical-context, making use of substantive words typical of the given sense in a window of several sentences; domain, concerned with the domain specificity of the used corpus and a disambiguation approach using this knowledge for the selection of senses. Depending on the data sources that are used for the disambiguation pipeline, it is not certain that topical-context or domain information will always be provided. Thus, in our work, we are interested in the context as asserted by micro-context, since it is easiest to obtain and, as Ide and Véronis (1998) also commented, highly informative in respect to the sense the target word is used with in the given surrounding.

In this paper, we investigate the effect of context window size on a machine learning system that makes use of memory-based learning (Daelemans and van den Bosch, 2005), explained further in section 3. As Daelemans and van den Bosch (2005) note, memory-based learning is highly sensitive to the amount of considered data in the form of features and their respective informativeness. Yet, Weaver (1955) claims that in order to disambiguate a given word, a wider context should be considered for the performance of the system to rise overall. However, a wider context implies more data and thus further features, which, as a whole, closes the circle of an endless loop over the trade-off between amount and informativeness of the used data.

Based on the presented problem, our assumption is that, for a memory-based learning approach, extending the context will lead to system performance improvement. Since the local context of a word, or its micro-context, has been the most often used source of information in the state-of-the-art word sense disambiguation approaches, we revise the findings in the field relevant to our work in section 2. Further, in section 3, we introduce the data that we employed in our study as well as the word sense disambiguation system that was developed specifically for this investigation arrangement. In section 4, we describe the experimental setup as well as the results we achieved and discuss the findings overall. In the last section of the paper, section 5, we sum up our investigation and review possible future directions.

2 Related Work

The optimal size of the context window that needs to be considered during memory-based WSD has been an important problem in the field for a long time (Wang, 2005). The diversity of algorithms

Proceedings of the Student Research Workshop associated with RANLP 2011, pages 91–96,
Hissar, Bulgaria, 13 September 2011.

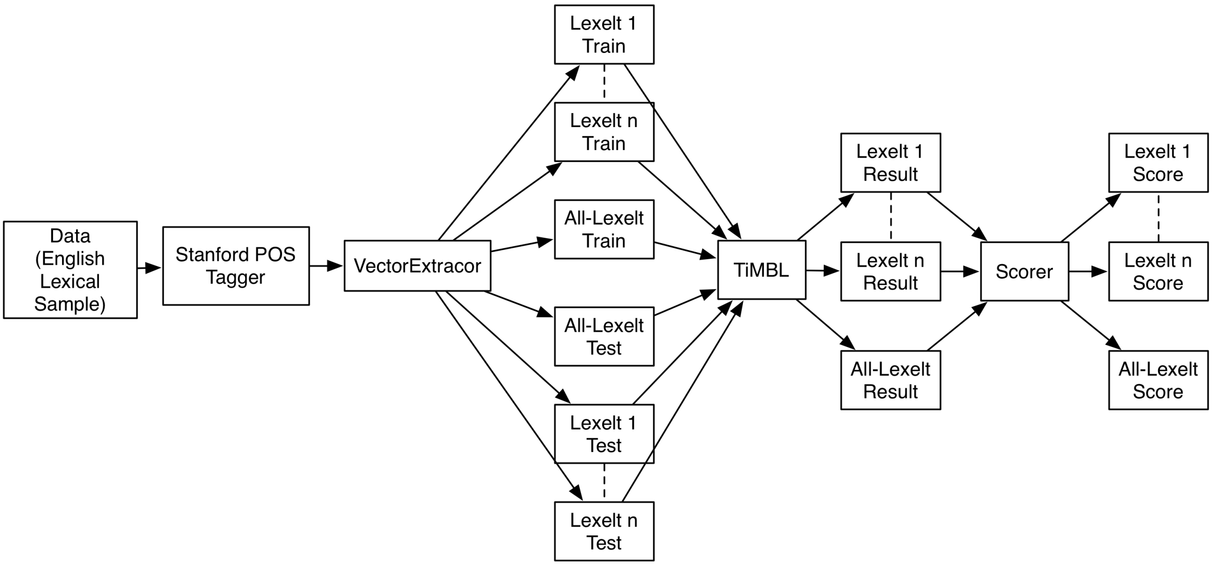

Figure 1: Overview of the WSD pipeline

used for the process, the data and ambiguity found in it, the language, that the final system is applied on as well as the variations in the distinct parameter optimization settings, constitute an immense pool of possibilities that can lead to a specific context window preference.

As Yarowsky and Florian (2002) find, the different methods and algorithms can benefit from the choice of the context size in a distinct way, which means that the optimal size of the micro-context can depend on the used method and that the selection of the size of the context leads to a variation of the WSD pipeline output. It was again Yarowsky (1994) that also claimed that the different types of ambiguity occurring in the data can be captured by a different size of the micro-context. In their work, Leacock et al. (1998) consider topical context as less informative than the immediate context around the target word. The authors look at various local context windows and suggest that a range of n=3 or n=4, meaning a context window of three, respectively four words before and after the target lemma, provide enough information from the local context. Based on his empirical study, Yarowsky (1994) also concludes that a window of 3 words around the target lemma leads to the optimal results. The latter findings became a default setup for multiple systems over the last few decades since a smaller context window is computationally more feasible than a bigger one (Decadt et al., 2004). Li et al. (2009), on the other side, use the Chinese Senseval[1] data set to look at a variation of the context window going beyond the idea of symmetric combination of lemmas before and after the target one.

Right in the beginning of machine translation, Weaver (1955) expressed a hope that not only the most optimal context window can be discovered but also the smallest one such that the correct sense of the target word is still selected. Yet, almost six decades later, there is still no specification of which size of the window needs to be used in which experimental setup. This provides a clear motivation for further investigation in the area.

3 The System

The data used in our research is retrieved from the WSD competition Senseval 3 (Mihalcea and Edmonds, 2004), namely the test and train files of the English Lexical Sample Task (Mihalcea et al., 2004). Lexical sample tasks use a small set of words and corpus instances of these words. Due to the reduced size of the data, a supervised machine-learning approach was applicable, in which we extract context information surrounding the ambiguous word.

The disambiguation pipeline (an overview of which is shown in figure 1) starts with a preparation process of the sentences, in which we tag every word with its part of speech (POS). This first pre-processing was carried out with Stan-

[1] http://www.senseval.org/senseval3/data.html

Feature	Description	Example
CT$_{-5}$	TP -5 from TW	,
CT$_{-4}$	TP -4 from TW	and
CT$_{-3}$	TP -3 from TW	I
CT$_{-2}$	TP -2 from TW	’d
CT$_{-1}$	TP -1 from TW	once
CT$_{0}$	TW	decided
CT$_{1}$	TP 1 from TW	to
CT$_{2}$	TP 2 from TW	wash
CT$_{3}$	TP 3 from TW	all
CT$_{4}$	TP 4 from TW	his
CT$_{5}$	TP 5 from TW	clothes
CP$_{-5}$	POS of TP -5 from TW	,
CP$_{-4}$	POS of TP -4 from TW	CC
CP$_{-3}$	POS of TP -3 from TW	PRP
CP$_{-2}$	POS of TP -2 from TW	MD
CP$_{-1}$	POS of TP -1 from TW	RB
CP$_{0}$	POS of target word	VBD
CP$_{1}$	POS of TP 1 from TW	TO
CP$_{2}$	POS of TP 2 from TW	VB
CP$_{3}$	POS of TP 3 from TW	PDT
CP$_{4}$	POS of TP 4 from TW	PRP$
CP$_{5}$	POS of TP 5 from TW	NNS
NA	first noun after TW	clothes
NB	first noun before TW	cleanliness
VA	first verb after TW	wash
VB	first verb before TW	had
PA	first preposition after TW	to
PB	first preposition before TW	for

Table 1: The pool of features used for classification (TP n is the token at position n and TW is the target word) and the values in the respective vector

ford’s POS-tagger[2] (Toutanova et al., 2003). After the original files are tagged, the output is further used from the next component in our WSD system, which extracts the desired information in the form of features building up a feature vector. It is also important to state, that the output of this second step is one separate file for each lexelt in the data, as well as one file containing data for all lexelts. A lexelt consists of a lemma and its word class. Out of a total of 57 lexelts, we then end up with 57 pairs of single-lexelt test and train files and 1 pair of all-lexelt test and train files.

In the extraction of vectors, we started with the feature set used in (Kübler and Zhekova, 2009), which is composed of tokens and POS-tags of the ambiguous word and its surrounding words, plus the first verb, noun and preposition before and after the ambiguous word, as shown in table 1.

For the actual classification process, we used TiMBL[3] (Daelemans et al., 2007), which is a considerably efficient decision-tree implementation of the k-nearest neighbor classification algorithm. We used the IB1-IG algorithm to process each of our train and test file pairs. We did not approach a parameter optimization, since we investigate the pure effect of the context window on the system performance and not the system’s best possible performance. Again the output of this step is a file for each lexelt as well as a combined file including all lexelts, with both the feature vectors from the test set and the newly added senses assigned to each of them. Our system transforms this output in the format needed by the scoring software integrated in the pipeline. For this purpose we used the scorer provided by Senseval 3. Additionally, to score each lexelt based on TiMBL’s prepared output file, an answer key file for each lexelt and a sense map (also provided in the data package from Senseval 3) were used.

4 Experiments

In order to investigate the actual effect of the micro-context on the IB1-IG algorithm, we approached several experiments, the setup of which we describe further in section 4.1. The results that we obtain are listed and discussed in section 4.2.

4.1 Experimental Setup

The features CT and CP are optimizable, in the sense that one can increase the value of n and expand the n-gram window to extract vectors. Thus, we approached altogether the following six experimental system runs: EX0 ($n = 0$), EX1 ($n = 1$), EX2 ($n = 2$), EX3 ($n = 3$), EX4 ($n = 4$) and EX5 ($n = 5$). In the EX0 setting, with $n = 0$, vectors are composed of CT0 and CP0 only (i.e no more than the target word itself is regarded) plus the non-optimizable features (NA,NB,VA,VB,PA and PB), for which the n value is irrelevant. With this initial feature set, we obtain the system performances, register them in a table and terminate EX0. For subsequent experiments we increment n simultaneously and symmetrically, i.e. for each $-x$ included (where x = some feature), a $+x$ is also included. In EX1, that results in the inclusion of the features CT+1, CT-1, CP+1 and CP-1 to the previous set used in EX0. The largest feature set in this study, namely that used in EX5 with $n = 5$, is demonstrated in table 1. This vector was extracted from the following corpus instance:
“...I had a mania for cleanliness, and I’d once decided to wash all his clothes...”.

Once we have extended the context until $n = 5$

[2] http://www-nlp.stanford.edu/software

[3] http://ilk.uvt.nl/timbl

POS	EX0		EX1		EX2		EX3		EX4		EX5	
	fine	coarse	fine	coarse	fine	coarse	fine	coarse	fine	coarse	fine	coarse
a	39.4	**55.1**	40.3	51.0	**41.9**	52.7	36.8	44.9	36.0	46.4	37.3	47.6
n	56.1	64.4	**59.1**	**67.1**	56.0	64.0	56.9	64.1	55.8	63.5	57.4	65.40
v	59.2	62.4	**64.6**	**68.0**	63.8	67.1	62.7	66.5	62.2	65.7	62.5	66.2

Table 2: System results across word types.

and obtained all results for each separate lexelt, for all lexelts together, and for every experiment, we analyze the evolution of performances. According to the assumption that more context yields better performance, we expected to conclude that:

- The larger the context-window we have in the system, the better the performance;
- From a certain point onwards, this gain is either irrelevant or reduces system performance, since too much information tends to mean more noise in the automated learning process.

4.2 Experimental Results

The scoring software provided by Senseval 3 allowed for the scoring step to be carried out in fine-grained scoring mode and in coarse-grained mode as well. The scores that we obtained are listed in table 3 in the form of the harmonic mean (F-score) of precision and recall for both modes. As *total* scores we report the average scores of all separate word experts and as *combined* we list the scores that the single classifier working with data for all words obtained. Table 2 offers an averaged performance of the system per word type. What the figures show is that for virtually all experiments (except partially in EX0), for both fine- and coarse-grained scores, ambiguity on verbs is better resolved than ambiguity on nouns with our system. The linguistic feature used in the vectors is mainly POS, which indicates that such a feature has more relevance in the disambiguation of verbs in comparison to other word types.

What we find is that there is a direct correlation between the amount of possible senses for the same word and the accuracy of the system. Words with 10 or more senses, for instance, had scores ranging from 20.8 to 80.6. In respect to words with 5 or less senses where scores ranged from 38.5 to 96.9. This is another indicator that factors as the feature set employed, the learning algorithm used as well as the level of ambiguity of the given word have a direct influence on the system performance.

There is a difference in performance when granularity is changed. Fine-grained scoring methods tend to lead to lower scores if compared to coarse-grained scores. In our case, regardless of context window size, coarse-grained scores were indeed higher than fine-grained scores for the same experiment. In both cases, fine- and coarse-grained, the context window from experiment 1 results in the majority of best scores. Nonetheless, it is important to note, that with the few lexelts for which larger context windows worked better, namely in 6 percent of the cases for window 3, and 10 percent for window 5, granularity does play a role. In those rare cases, context window 3 displayed the best average performance for a fine-grained disambiguation. Whereas window 5 functioned best for a coarse-grained disambiguation. This suggests, that in case the lexelt scores are not sufficient a context window of n $\geq$3 can be considered.

A similar irregularity in performance gain or loss between fine- and coarse-grained disambiguation is visible for one specific word class, namely adjectives as shown in table 2. We should, however, note that the data for adjectives is not representative enough, since we only have 5 instances and, thus, more data is needed in order to be sure that this variation is indeed relevant.

Comparing the *total* and *combined* performance, it is surprising to see that the classifier that was trained on the whole data set performed better in all experimental settings and modes than the average performance of all separate word experts. We believe this is due to the fact that TiMBL uses an information gain algorithm, allowing it to evaluate features better if it has more data. This implies that in a setting, in which no per word classifier optimization is approached, a single classifier is indeed sufficient.

Overall, we see that the context window size used in EX1, which is $n = 1$, results in the best performance in the majority of cases. This leads to the fact that, in the simple setting of our system, a micro-context of one word before and after the target one performs best. Our findings then contradict our expectations in the sense that we estimated an increase of system performance with a context window of 3 or 4, since such a distance has been common practice in the area for the last few decades and was proposed as optimal by Leacock et al. (1998). A possible explanation for this outcome stands behind the features extracted from a wider context, which can bring more noise than helpful information for a memory-based classifier.

5 Conclusion and Future Work

We have shown that the default size for a context window in memory-based word sense disambiguation, used for more than a decade, is hardly still optimal. We now find that a window of ± 1 yields the best possible averaged results over all ambiguous words. This work also raised some issues which should be investigated further, for instance why the disambiguation of certain words works so much better with bigger windows. Another point of interest is the fact that the average score achieved from all separate word experts is lower than the score achieved from the classifier working with all lexelts simultaneously. Lastly, it might be beneficial to investigate if these findings hold true for more than just the English language. As R.Martin (1992) has indicated, English tends to keep relevant context information very close to the word in question, which would explain why our small window of ± 1 worked so well.

References

Walter Daelemans and Antal van den Bosch. 2005. *Memory Based Language Processing*. Cambridge University Press.

Walter Daelemans, Jakub Zavrel, Ko van der Sloot, and Antal van den Bosch. 2007. TiMBL: Tilburg Memory Based Learner – version 6.1 – Reference Guide. Technical Report ILK 07-07, Induction of Linguistic Knowledge, Computational Linguistics, Tilburg University.

Bart Decadt, Véronique Hoste, Walter Daelemans, and Antal V. den Bosch. 2004. In Rada Mihalcea and Phil Edmonds, editors, *Proceedings of the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text (Senseval-3)*, pages 108–112.

Nancy Ide and Jean Véronis. 1998. Introduction to the Special Issue on Word Sense Disambiguation: The State of the Art. *Computational Linguistics*, 24(1):1–40.

Sandra Kübler and Desislava Zhekova. 2009. Semi-Supervised Learning for Word Sense Disambiguation: Quality vs. Quantity. In *Proceedings of the International Conference RANLP-2009*, pages 197–202, Borovets, Bulgaria, September. ACL.

Claudia Leacock, George A. Miller, and Martin Chodorow. 1998. Using corpus statistics and WordNet relations for sense identification. *Computational Linguistics*, 24:147–165, March.

Gang Li, Guangzeng Kou, Ercui Zhou, and Ling Zhang. 2009. Symmetric Trends: Optimal Local Context Window in Chinese Word Sense Disambiguation. In *Proceedings of the 2009 Ninth International Conference on Hybrid Intelligent Systems - Volume 03*, HIS '09, pages 151–154, Washington, DC, USA. IEEE Computer Society.

Rada Mihalcea and Philip Edmonds, editors. 2004. *Proceedings of Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, Barcelona, Spain.

Rada Mihalcea, Timothy Chklovski, and Adam Kilgarriff. 2004. The SENSEVAL–3 English Lexical Sample Task. In *SENSEVAL-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 25–28, Barcelona, Spain. Association for Computational Linguistics.

James R.Martin. 1992. *English text: system and structure*. Benjamins, Philadelphia [u.a.]. XIV, 620 S. : graph. Darst.

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 HLT-NAACL*, pages 252–259.

Xiaojie Wang. 2005. Robust utilization of context in word sense disambiguation. In *CONTEXT*, pages 529–541.

Warren Weaver. 1955. Translation. In William N. Locke and A. Donald Boothe, editors, *Machine Translation of Languages*, pages 15–23. MIT Press, Cambridge, MA. Reprinted from a memorandum written by Weaver in 1949.

David Yarowsky and Radu Florian. 2002. Evaluating sense disambiguation across diverse parameter spaces. *Natural Language Engineering*, 8:293–310.

David Yarowsky. 1994. Decision lists for lexical ambiguity resolution: application to accent restoration in Spanish and French. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, ACL '94, pages 88–95, Stroudsburg, PA, USA. Association for Computational Linguistics.

LEXELT	POS	senses	EX0		EX1		EX2		EX3		EX4		EX5	
			fine	coarse	fine	coarse	fine	coarse	fine	coarse	fine	coarse	fine	coarse
activate	v	5	77.4	77.4	71.0	71.0	68.8	68.8	63.4	63.4	63.4	63.4	65.6	65.6
add	v	6	54.9	54.9	78.9	78.9	81.5	81.5	79.8	79.8	80.6	80.6	78.0	78.0
appear	v	3	66.9	66.9	71.0	71.0	68.5	68.5	69.4	69.4	70.2	70.2	70.2	70.2
argument	n	5	44.7	55.2	47.9	52.1	39.4	44.8	44.7	51.0	46.8	52.1	51.1	56.2
arm	n	6	84.7	84.7	87.0	87.0	84.7	84.7	83.2	83.2	83.2	83.2	82.4	82.4
ask	v	6	32.9	32.9	59.9	59.9	55.9	55.9	55.9	55.9	51.9	51.9	51.9	51.9
atmosphere	n	6	56.7	54.8	55.0	53.2	40.0	40.3	43.3	43.5	41.7	41.9	43.3	43.5
audience	n	4	69.1	97.4	70.2	96.4	71.3	96.4	72.3	95.4	69.1	95.4	71.3	96.4
bank	n	10	72.2	79.1	68.3	78.3	70.6	79.1	71.4	78.3	73.8	80.6	67.5	79.1
begin	v	4	47.9	47.9	51.1	51.1	51.1	51.1	52.7	52.7	44.7	44.7	44.7	44.7
climb	v	5	44.5	44.5	56.9	56.9	59.9	59.9	59.9	59.9	61.5	61.5	61.5	61.5
decide	v	4	73.8	73.8	72.1	72.1	70.5	70.5	70.5	70.5	75.4	75.4	77.0	77.0
degree	n	7	62.9	78.6	68.1	82.5	70.7	85.7	68.1	82.5	69.0	82.5	65.5	81.0
difference	n	5	54.1	63.4	55.1	60.4	48.0	53.5	48.0	54.5	44.9	53.5	48.0	55.4
different	a	5	45.8	62.0	47.9	62.0	45.8	62.0	41.7	56.0	47.9	62.0	41.7	60.0
difficulty	n	4	39.1	87.0	52.2	87.0	43.5	82.6	47.8	82.6	39.1	87.0	47.8	87.0
disc	n	4	39.6	39.6	45.1	45.1	44.0	44.0	40.7	40.7	38.5	38.5	40.7	40.7
eat	v	7	83.5	83.5	87.1	87.1	82.4	82.4	75.3	75.3	76.5	76.5	76.5	76.5
encounter	v	4	58.5	93.8	55.4	96.9	58.5	96.9	60.0	96.9	61.5	96.9	60.0	96.9
expect	v	3	65.2	65.2	71.0	71.0	75.4	75.4	75.4	75.4	75.4	75.4	73.9	73.9
express	v	4	54.7	61.8	56.6	65.5	54.7	61.8	49.1	61.8	50.9	60.0	56.6	67.3
hear	v	7	43.8	50.0	53.1	59.4	56.2	62.5	56.2	62.5	53.1	59.4	53.1	59.4
hot	a	22	71.4	71.4	78.6	78.6	78.6	78.6	71.4	71.4	71.4	71.4	69.0	69.0
image	n	7	58.9	58.9	50.7	50.7	49.3	49.3	50.7	50.7	52.1	52.1	54.8	54.8
important	a	5	38.5	66.7	23.1	46.7	30.8	53.3	23.1	33.3	23.1	33.3	15.4	33.3
interest	n	7	69.2	69.6	70.3	70.7	68.1	68.5	69.2	69.6	65.9	66.3	70.3	69.6
judgment	n	7	34.4	40.6	40.6	46.9	53.1	56.2	53.1	53.1	53.1	53.1	53.1	56.2
lose	v	9	47.2	47.2	36.1	36.1	47.2	47.2	33.3	33.3	33.3	33.3	33.3	33.3
mean	v	7	50.0	50.0	70.0	70.0	67.5	67.5	72.5	72.5	70.0	70.0	70.0	70.0
miss	v	8	40.0	40.0	50.0	50.0	43.3	43.3	46.7	46.7	43.3	43.3	43.3	43.3
note	v	3	60.6	60.6	60.6	60.6	60.6	60.6	62.1	62.1	60.6	60.6	63.6	63.6
operate	v	5	44.4	55.6	72.2	88.9	72.2	77.8	66.7	83.3	61.1	77.8	61.1	77.8
organization	n	7	69.1	76.8	67.3	78.6	72.7	83.9	78.2	89.3	70.9	85.7	76.4	92.9
paper	n	7	43.9	51.4	45.9	59.5	38.8	50.5	38.8	51.4	39.8	52.3	41.8	53.2
party	n	5	63.6	63.6	64.5	64.5	64.5	65.4	63.6	64.5	64.5	64.5	62.6	63.6
performance	n	5	25.3	41.2	28.9	44.7	28.9	44.7	33.7	44.7	31.3	37.6	31.3	36.5
plan	n	3	76.8	77.8	73.9	76.4	72.5	75.0	72.5	75.0	75.4	75.0	81.2	81.9
play	v	11	44.2	44.2	42.3	42.3	38.5	38.5	42.3	42.3	32.7	32.7	38.5	38.5
produce	v	6	55.9	57.4	55.9	57.4	55.9	57.4	50.5	51.1	51.6	54.3	50.5	53.2
provide	v	6	85.1	92.8	85.1	94.2	88.1	95.7	95.5	98.6	95.5	98.6	95.5	97.1
receive	v	9	85.2	85.2	88.9	88.9	81.5	81.5	88.9	88.9	85.2	85.2	88.9	88.9
remain	v	3	82.6	82.6	82.6	82.6	87.0	87.0	89.9	89.9	88.4	88.4	89.9	89.9
rule	v	5	60.0	60.0	66.7	66.7	56.7	56.7	60.0	60.0	63.3	63.3	56.7	56.7
shelter	n	4	49.4	49.4	60.5	60.5	51.9	51.9	48.1	48.1	51.9	51.9	55.6	55.6
simple	a	7	25.0	58.8	31.2	47.1	37.5	52.9	31.2	47.1	25.0	52.9	43.8	58.8
smell	v	7	55.6	57.4	66.7	70.4	64.8	68.5	70.4	74.1	66.7	70.4	70.4	72.2
solid	a	14	16.1	16.7	20.8	20.8	16.7	16.7	16.7	16.7	12.5	12.5	16.7	16.7
sort	n	4	59.0	67.9	71.1	85.7	66.3	78.6	66.3	78.6	67.5	79.8	66.3	79.8
source	n	6	48.3	51.7	58.6	62.1	41.4	44.8	44.8	44.8	37.9	37.9	37.9	41.4
suspend	v	7	53.1	53.1	45.3	45.3	45.3	45.3	51.6	51.6	50.0	50.0	46.9	46.9
talk	v	9	67.1	67.1	72.6	72.6	76.7	76.7	72.6	72.6	71.2	71.2	71.2	71.2
treat	v	9	56.1	61.4	56.1	57.9	56.1	57.9	45.6	45.6	45.6	47.4	50.9	52.6
use	v	5	71.4	71.4	78.6	78.6	71.4	71.4	64.3	64.3	64.3	64.3	71.4	71.4
wash	v	12	67.6	73.5	64.7	70.6	58.8	76.5	58.8	79.4	55.9	76.5	52.9	76.5
watch	v	7	60.8	80.4	82.4	88.2	78.4	86.3	74.5	86.3	72.5	82.4	72.5	82.4
win	v	7	59.0	61.5	51.3	56.4	56.4	64.1	43.6	53.8	56.4	61.5	51.3	56.4
write	v	8	43.5	43.5	56.5	56.5	52.2	52.2	47.8	47.8	56.5	56.5	52.2	52.2
total			56.3	62.5	60.6	66.2	59.1	64.7	58.4	63.7	57.6	63.3	58.5	64.2
combined			59.2	64.2	61.9	66.9	61.0	65.8	61.6	66.2	60.6	65.1	60.3	65.0

Table 3: System results.