A Methodology for Word Sense Disambiguation at 90% based on large-scale CrowdSourcing

Oier Lopez de Lacalle
University of the Basque Country
oier.lopezdelacalle@ehu.eus

Eneko Agirre
University of the Basque Country
e.agirre@ehu.eus

Abstract

Word Sense Disambiguation has been stuck for many years. In this paper we explore the use of large-scale crowdsourcing to cluster senses that are often confused by non-expert annotators. We show that we can increase performance at will: our in-domain experiment involving 45 highly polysemous nouns, verbs and adjective (9.8 senses on average), yields an average accuracy of 92.6 using a supervised classifier for an average polysemy of 6.1. Our proposal has the advantage of being cost-effective and being able to produce different levels of granularity. Our analysis shows that the error reduction with respect to fine-grained senses is higher, and manual inspection show that the clusters are sensible when compared to those of OntoNotes and WordNet Supersenses.

1 Introduction

Word sense ambiguity is a major hurdle for accurate information extraction, summarization and machine translation. The utility of Word Sense Disambiguation (WSD) depends on the accuracy and on how useful the sense distinctions are. The first issue is quantitative, as it can be measured using a WSD system on certain dataset. The second examines whether the sense distinctions are appropriate, which varies from application to application. Although usefulness can be explored in a downstream application (Agirre et al., 2008), it is usually assessed subjectively, discussing the quality of the sense distinctions (Palmer et al., 2007). Both issues (performance and usefulness) are linked to the granularity of the sense inventory, and conflict with each other: finer granularity might produce more useful distinctions but the accuracy would be worse, and vice-versa.

WordNet (Fellbaum, 1998) is the most widely used resource to build word sense disambiguation tools and word sense annotated corpora, including recent large efforts (Passonneau et al., 2012), but its fine-grainedness has been mentioned to be a problem (Hovy et al., 2006; Palmer et al., 2007).

We think that a desiderata for a sense inventory would be that it provides useful sense distinctions and useful performance across a large range of applications. We would also add that it should be tightly integrated with WordNet, given its prevalence on NLP applications, and we thus focus on sense inventories which are mapped to WordNet.

In order to asses usefulness, we need specific measures. Downstream application is difficult, and unfeasible for new proposals, as a full-fledged sense inventories and associated annotations are necessary. We can instead estimate usefulness of proposed sense inventories using several proxy measures:

- High **polysemy**. Note that polysemy alone could be misleading, as a word with many senses might be skewed to a single sense: 99% of occurrences could belong to a single sense, while the rest are only seen once. Besides the absolute polysemy, we can use the accuracy of the **most frequent sense** (MFS, estimated in train data and applied to test data) as a simple

*Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics (*SEM 2015)*, pages 61–70,
Denver, Colorado, June 4–5, 2015.

and effective indication of skewness. High polysemy and low MFS are desirable properties.

- High performance, as measured the accuracy of a supervised system trained on hand-annotated data. The higher the **accuracy**, the better.

- Flexible sense granularity, that is, the ability to produce different degrees of polysemy and accuracy, from fine-grained to coarse-grained. When comparing sense inventories with different granularities, absolute MFS and supervised performance are not enough. We propose to use **error reduction** of the supervised system with respect to the MFS as a measure of the balance between low MFS and high supervised performance. The larger the error reduction the better.

- Manual inspection of the sense distinctions, as a complement to quantitative measures.

We propose to use crowdsourced annotations (Passonneau and Carpenter, 2014) to cluster WordNet senses that are often confused by non-expert annotators. Our method can provide clusters at different levels of granularity. We show that we can construct clusters yielding around 90% accuracy for 45 words, with higher error reduction with respect to MFS than fine-grained senses. By construction, we merge senses which are often confused by annotators, yielding sensible sense clusters, as corroborated by manual inspection.

The paper is structured as follows. Section 2 mentions related work. We then present the annotations, followed by the clustering procedure. Section 5 report the main experiments. Section 6 compares our clustering to that of OntoNotes followed by a comparison to WordNet Supersenses. Section 8 draws the conclusions.

2 Related Work

Our work is close to (Passonneau and Carpenter, 2014) in that we use the same dataset and annotations presented in that work. They present a comparison of conventional expert-guided annotation model with a probabilistic annotation model that does not take agreement into account.

Previous efforts to cluster WordNet senses in order to produce coarse-grained inventories have shown that improved results can be obtained, but we think our approach fits the desiderata better. For instance, clustering together senses which have the same Semantic File (also called Supersenses) allowed the best supervised WSD system to date (Zhong and Ng, 2010) to increase accuracy from 58.3% to 82.6% in the Semeval 2007 all-words dataset (Navigli et al., 2007). Semantic Files are useful, but don't allow to provide flexible sense inventories.

The OntoNotes project (Hovy et al., 2006) devised a manual grouping method which explicitly sought 90% accuracy. Although the method was shown to be successful, the fixed sense groupings had to be produced manually, included complex mappings to WordNet (cf. Section 6), and was a limited exercise, with annotations for around 4900 words. Our work is similar in spirit to OntoNotes, but use a different methodology which allows for flexible granularity, as the annotation is done at the fine-grained level, and the clustering is done later fully automatically.

Automatic clustering algorithms are not new. (Tou Ng et al., 1999) propose to use annotator agreement to cluster senses, reporting higher inter-annotator agreement after clustering. We are in part inspired by their approach, as we extend it from two annotators to a sample with 25 annotators, and validate the approach with WSD systems.

The rest of approaches use other sources of information. Peters et al. (1998) make use of the WordNet hierarchy to group close senses. Mihalcea and Moldovan (1999) present similar approach that is based in the structure of WordNet. Tomuro (2001) presents are more principled algorithm based on *Minimum Description Length*. A work which is closely related to our work is (Agirre and Lopez de Lacalle, 2003), in which they examine a variety of information sources to cluster WordNet word senses, including a hierarchical clustering based on distributional information. Snow et al. (2007) present a supervised learning algorithm that learns merging senses and make use of wide range of WordNet-based and corpus-based features. (Navigli et al., 2007) mapped WordNet to the top level sense distinctions in the Oxford Dictionary of

English. All the above rely on automatic measures, while our method is based on human annotations.

3 MASC Crowdsourced annotations

The corpus used in the experiments is part of the Manually Annotated Sub-Corpus of the Open American National Corpus, which contains a subsidiary word sense sentence corpus consisting of approximately one thousand sentences per word annotated with WordNet 3.0 sense labels (Passonneau et al., 2012). In this work we make use of a publicly available subset of 45 words (17 nouns, 19 verbs and 9 adjective, see Table2) that have been annotated, 1000 sentences per target word, using crowdsourcing (Passonneau and Carpenter, 2014). The authors collected between 20 and 25 labels for every sentence. They showed that a probabilistic annotation model based on crowdsourced data was effective, with favorable quality when compared to a conventional expert-guided annotation model.

4 Clustering Procedure

Having access to multiple annotations of the same item allows to identify correlations among senses of a word. In particular, we can mine how many times the annotators confused 2 particular senses of a word. If two senses are confused very often, it will signal that the annotators find the differences between the two senses difficult to discriminate in context. We also want to note that, in some cases, the context might be underspecified, and several senses might hold at the same time. We left this second phenomena for a future study.

We built a confusion matrix for each target word counting how many times two distinct senses are annotated in the same instance. More formally, the confusion of two senses of a target word $\text{conf}(s_1, s_2)$ is defined as follows:

$$\frac{1}{I}\sum_{i=1}^{I}\frac{1}{\binom{J_i}{2}}\sum_{m=1}^{J_i-1}\sum_{n=m+1}^{J_i}\mathbb{I}((y_{i,n}=s_1 \wedge y_{i,m}=s_2)\vee (y_{i,n}=s_2 \wedge y_{i,m}=s_1))$$

where I is number of instances of the word, J_i is the number of turkers that annotated instance i, and $y_{i,m}$ is the annotation of turker m in instance i. Finally, $\mathbb{I}(s) = 1$ iff the condition expressed in s is true.

We cluster the senses based on the information in the confusion matrix, i.e. two senses (s_1, s_2) will tend to be in the same cluster if $\text{conf}(s_1, s_2)$ is high. We used agglomerative hierarchical clustering for the sake of simplicity, as we obtain one hierarchy of senses in one go, and then used different cuts in the hierarchy to obtain clusters of different sense granularities.

In order to obtain the target coarse-grained inventory, the procedure was the following: (0) we start at the leaves of the hierarchy, that is, with the fine-grained senses; (1) we train and test a word sense disambiguation algorithm on development data using the current sense distinctions (see the next Section for details); (2) if the accuracy is higher than 90%, or if there are only two senses left, we stop and output the current sense distinctions; (3) we go up one level in the hierarchical cluster, joining together the two senses with highest confusion score, and go to step (1). Note that the algorithm does not guarantee obtaining 90% on the training data. Once the coarse-grained senses are obtained, we train the word sense disambiguation on the development data and test over held-out data, yielding the final accuracy scores.

In order to contrast results, we also produced hierarchies of senses based on random clustering, where the clusters yield the same sense granularity as those of the confusion-based clustering explained above. We produced 10 random clustering for each word, and averaged over the runs to obtain the final accuracy.

5 Experiments

The gold standard is based on the multiple annotations in the corpus, but a single sense was selected as the correct one, following (Passonneau and Carpenter, 2014), which use a probabilistic annotation model (Dawid and Skene, 1979). We split the 1000 examples for each word into development and test, sampling 85% (and 15% respectively) at random, preserving the overall sense distribution.

The Word Sense Disambiguation algorithm of choice is *It Make Sense* (IMS) (Zhong and Ng, 2010), which reports the best WSD results to date. IMS is a freely available Java implementation[1],

[1] http://www.comp.nus.edu.sg/~nlp/

which provides an extensible and flexible platform for researchers interested in using a WSD component. Following Lee and Ng (2002), IMS adopts support vector machines as the classifier and integrates the state of the features extractors including parts-of-speech of the surrounding words, bag of words features, and local collocations as features.

IMS provides ready-to-use models trained with examples collected from parallel texts , SEMCOR (Miller et al., 1993), and the DSO corpus (Ng and Lee, 1996). In our experiments we train IMS with the train examples of the crowdsourced MASC. We used IMS out-of-the-box, using the default parametrization and built-in feature extraction. We compare results obtained with IMS against the Most Frequent Sense (MFS), which was estimated using the training corpus. Both systems (IMS and MFS) could be trained on fine-grained senses, on coarse-grained senses induced from the confusion matrix using the 90% threshold described above ($\text{Coarse}_{\text{conf}}$) and coarse-grained senses induced from random clustering using the 90% threshold ($\text{Coarse}_{\text{random}}$). We also used sense clusters from OntoNotes and WordNet Supersenses (cf. Sections 6 and 7) .

5.1 Main results

The results of the six systems on development and test data are shown in Table 1, showing that we successfully attained an accuracy over 90% on average. The results for random clustering show that not any clustering yields meaningful results. Due to variation of the random sense-hierarchies, we calculated the upper and lower margins with 95% of confidence level (79.2-80.0 accuracy in test). The results show that random clustering performs significantly lower than the confusion based clustering. The results in development and test are very similar, confirming that the confusion information is stable in our in-domain scenario.

All in all, as Table 2 shows, 30 words out of the 45 attain an accuracy higher than 90% in test (14 out of 17 nouns, 11 out of 19 verbs and 5 out of 9 adjectives). The precision for the words which do not attain 90% is 87.4% on average, and 85.4% for adjective, being the lowest. The polysemy is reduced from 9.8 to 6.2.The appendix shows detailed information for each target word. In all but 3 words coarse-grained accuracy is above fine-grained. Note that MFS and IMS produce the same results in 11 words out of 45. We will revisit MFS in Section 6.

Figure 1 plots, on top, the results (on test) grouped on MFS and IMS for easier comparison. The figure also plots the error reduction of each coarse-grained inventory with respect to fine-grained. The higher

software.html

	Development		Test	
	MFS	**IMS**	**MFS**	**IMS**
Fine-grained	47.2	73.2	46.2	73.1
$\text{Coarse}_{\text{random}}$	60.4	79.9	60.2	79.6
$\text{Coarse}_{\text{conf}}$	84.2	92.9	84.1	92.6

Table 1: Development and test results using cross-validation (left side) and test results (right side) for IMS and MFS using three sense inventories.

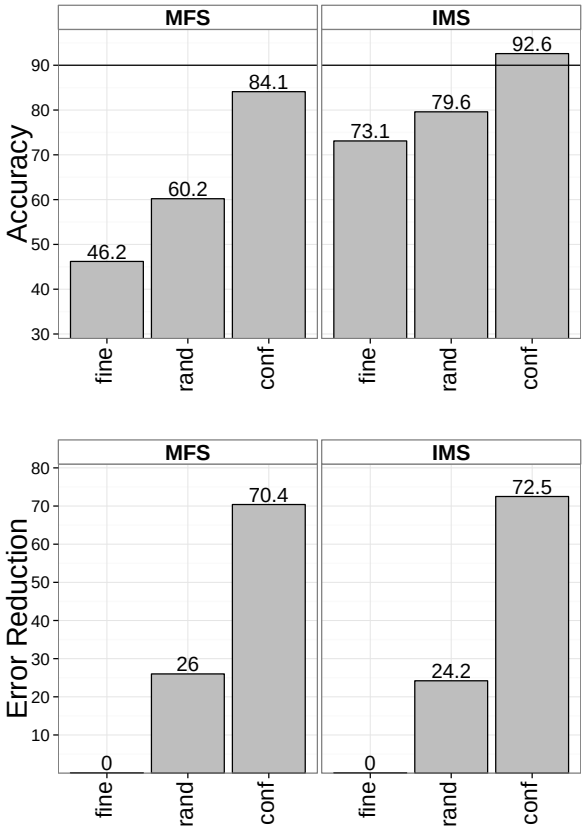

Figure 1: Accuracy on test of sense granularities and methods (top) and error reduction with respect to fine-grained (bottom), for the three sense inventories.

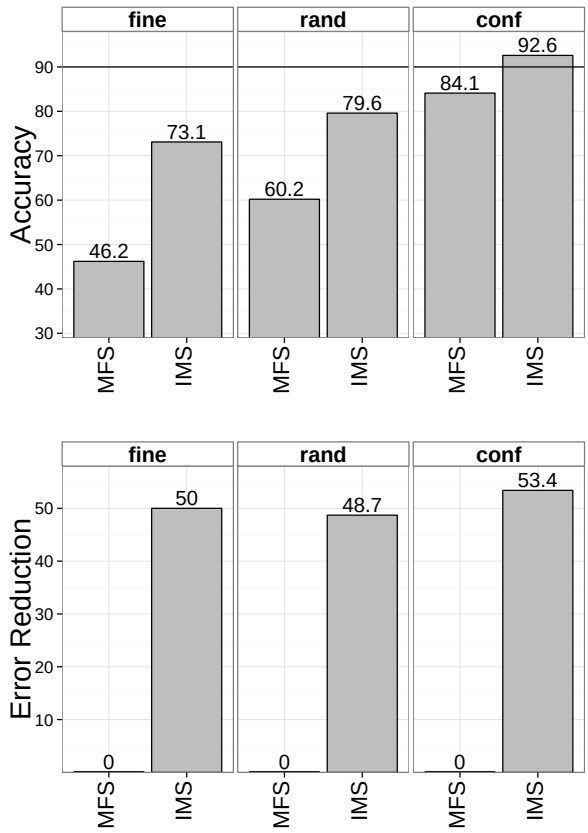

Figure 2: Accuracy of sense granularities and methods (top) and error reduction of IMS with respect to MFS (bottom), for the three sense inventories.

error reduction of our coarse-grained inventory with respect to the random clustering shows that the **clusters are meaningful**, and that the performance gain is not an artifact of reducing the sense inventory[2].

Figure 2 plots, on top, the results (on test) grouped on each sense inventory. The figure also plots the error reduction of IMS with respect to MFS in each coarse-grained inventory. The better error reduction of IMS with respect to MFS for our coarse-grained inventory shows that **our clusters are easier to learn**, in that reducing the sense inventory increases the delta with respect to the MFS baseline. Note that reducing the sense inventory is not enough to show this effect, as exemplified by the fact that the error reduction for the random clusters is lower than for the fine-grained senses.

[2]Note that, by construction, $\text{Coarse}_{\text{random}}$ and $\text{Coarse}_{\text{conf}}$ have the same granularity.

5.2 Flexible clustering

As we reached 90% of accuracy with relatively high polysemy, we also checked whether MFS could reach 90% of accuracy if we continued to cluster senses. The experiments in development confirmed that MFS gets above 90% at expenses of coarser grained senses than IMS does: On average, the fine-grained polysemy (9.8) would drop to 4.4, compared to the 6.2 when clustering to reach 90% using IMS. When we obtain MFS>90% 17 words have 2 senses and 40 words reach to 90% of accuracy, whilst when IMS>90% only 5 words are reduced to 2 senses and 42 words reach to 90%. This shows that it makes more sense to cluster senses using the performance of IMS as stopping criteria, as the polysemy is preserved better.

In case we continued clustering senses until we have 2 senses for each word, IMS would reach 98.2% and MFS 95.7%, with an error reduction of 58% over the MFS. Note that this error reduction compares favorably to that of our clustering when stopping at IMS>90%, showing that we could have kept clustering senses further without losing predictive power. These figures show that we could stop at arbitrary performance figures at the cost of obtaining highly skewed clusters (indicated by the high MFS value). We will revisit high MFS in Section 6.

6 Comparison to OntoNotes senses

In order to perform a qualitative study and check whether our sense clusters make sense, we decided to compare them to another coarse sense inventory which is mapped to WordNet. We chose Ontonotes 5.0 (Hovy et al., 2006), which also had the goal of attaining 90% sense accuracy. Alternatively, we could have used the Oxford Dictionary of English, which was mapped automatically to WordNet 2.1 (Navigli et al., 2007) but we preferred to factor out automatic mappings and version differences from the analysis.

Ontonotes contains lexical entries for 35 of our target words. The relation between the sense inventory of WordNet and OntoNotes is complex. Given that our work clusters WordNet senses, we focused on the 18 words where the OntoNotes senses where composed of one or several WordNet senses and where all WordNet senses were covered[3]. Table 2

[3]The rest of words include senses not mapped to Word-

	Fine-grained			Conf			Random		ON			SS		
Word	#S	IMS	Mfs	#S	IMS	Mfs	IMS	Mfs	#S	IMS	Mfs	#S	IMS	Mfs
common-j	10	70.5	39.0	7	87.0	83.6	74.3	46.9	-	-	-	1	100	100
fair-j	11	88.8	67.3	11	88.8	67.3	88.8	67.3	-	-	-	2	95.9	93.9
full-j	9	78.5	45.1	8	92.4	66.7	80.2	54.2	-	-	-	2	98.6	98.6
high-j	8	86.3	71.9	7	94.5	86.3	87.1	74.6	-	-	-	2	97.9	92.5
late-j	8	84.9	52.1	7	87.0	70.5	87.5	58.5	-	-	-	2	98.6	98.6
long-j	10	78.0	49.3	6	98.7	98.7	79.8	53.6	-	-	-	2	90.0	86.7
normal-j	5	81.9	66.0	4	85.4	79.9	83.8	72.2	-	-	-	1	100	100
particular-j	7	85.0	51.0	6	94.6	66.7	89.5	60.0	-	-	-	1	99.3	99.3
poor-j	6	76.2	52.4	2	94.6	62.6	88.2	73.6	-	-	-	2	100	100
board-n	10	88.9	79.9	9	93.8	93.1	89.8	80.9	7	88.9	79.9	5	89.6	79.9
book-n	12	56.3	64.4	11	88.9	89.6	58.3	65.3	-	-	-	5	65.9	71.1
color-n	9	65.5	32.4	2	99.3	99.3	85.6	76.8	-	-	-	4	84.8	73.8
control-n	12	79.5	46.6	6	96.6	93.8	82.8	52.0	-	-	-	8	78.1	46.6
date-n	9	80.1	24.1	4	91.5	65.2	84.2	43.4	-	-	-	5	91.5	85.1
family-n	9	64.3	26.6	2	100	100	79.7	64.9	-	-	-	3	99.3	99.3
image-n	10	70.6	49.0	7	90.9	85.3	74.6	59.0	-	-	-	7	77.6	60.8
land-n	12	57.6	20.8	6	96.5	92.4	62.9	43.2	-	-	-	6	62.5	28.5
level-n	9	69.9	52.1	7	94.5	94.5	74.6	59.3	7	85.6	77.4	5	77.4	53.4
life-n	15	58.0	21.7	5	89.5	88.8	67.3	43.8	-	-	-	10	65.0	46.2
number-n	12	87.7	71.2	11	92.5	86.3	89.6	73.9	-	-	-	5	89.0	71.2
paper-n	8	76.4	41.0	2	100	100	87.7	72.4	-	-	-	5	83.3	74.3
sense-n	6	93.8	38.6	6	93.8	38.6	93.8	38.6	6	93.8	38.6	3	94.5	66.9
time-n	11	90.1	48.4	7	94.5	93.4	89.7	51.7	-	-	-	5	92.9	48.4
way-n	13	72.1	55.8	7	91.2	78.9	78.6	62.4	-	-	-	8	78.2	59.9
window-n	9	75.9	38.6	3	91.7	60.7	84.9	59.3	-	-	-	4	90.3	62.8
work-n	8	69.8	20.5	2	85.9	79.5	80.9	63.1	-	-	-	5	75.1	38.5
add-v	7	40.3	49.3	2	91.0	91.0	74.3	77.1	3	90.3	90.3	6	40.3	49.3
appear-v	8	64.4	47.3	5	87.7	63.0	69.2	59.6	5	87.7	63.0	5	87.7	63.0
ask-v	8	78.3	36.4	6	96.5	96.5	83.9	53.8	-	-	-	3	100	100
find-v	17	62.4	28.4	13	86.5	85.8	65.7	30.8	6	80.9	58.9	7	75.2	41.8
fold-v	6	93.2	83.7	6	93.2	83.7	93.2	83.7	5	93.2	83.7	4	94.6	83.7
help-v	9	61.2	36.0	6	99.3	97.1	69.3	48.7	3	100	97.8	4	73.4	59.7
kill-v	16	63.9	59.7	11	89.6	86.8	67.0	62.5	9	89.6	86.8	7	82.6	81.2
know-v	12	63.1	35.4	7	89.2	77.9	64.8	40.3	7	81.5	48.7	2	100	100
live-v	8	73.5	47.6	3	97.3	94.6	82.5	66.9	-	-	-	3	91.2	87.8
lose-v	12	64.4	50.7	3	93.8	93.8	76.6	69.1	6	70.5	58.2	7	75.3	64.4
meet-v	14	69.7	28.9	2	86.6	59.9	83.5	61.1	7	82.4	52.1	8	78.9	59.9
read-v	12	82.8	73.9	11	85.1	80.6	84.5	76.1	8	82.1	75.4	4	91.0	88.8
say-v	12	64.9	35.7	7	96.1	96.1	67.5	44.0	6	96.1	92.2	3	100	100
serve-v	16	73.1	40.7	11	88.3	83.4	76.4	46.1	7	80.0	49.7	6	81.4	65.5
show-v	13	75.7	27.5	9	94.2	93.7	79.1	41.6	-	-	-	5	81.5	32.8
suggest-v	5	74.3	63.5	3	97.3	97.3	81.0	73.3	3	91.2	81.8	1	99.3	99.3
tell-v	9	57.6	38.9	6	86.1	83.3	69.2	54.6	4	94.4	92.4	3	97.9	97.2
wait-v	5	70.2	36.6	3	96.2	92.4	82.2	66.1	3	96.2	92.4	4	76.3	64.9
win-v	5	72.81	60.5	2	100	99.3	87.9	82.3	-	-	-	4	76.9	70.1
AVG 45 words	9.8	73.1	46.2	6.2	92.6	84.1	79.6	60.2	-	-	-	4.7	84.2	73.3
AVG 18 words	10.2	69.9	48.8	6.0	91.5	83.2	76.8	59.6	5.7	88.0	73.2	4.3	86.2	74.3

Table 2: The 45 words, with PoS, polysemy, IMS and Mfs accuracy for fine-grained, our clustering (Conf.), random clustering, OntoNotes coarse-grained senses (ON, cf. Section 6) and Supersenses (SS, cf. Section 7). The bottom rows report averages for the 45 words and the 18 words in OntoNotes.

Net, or cases where one WordNet sense was mapped to several OntoNotes senses

Conf	ON	WN	#	Gloss
1	1	1	421	Make an addition (to); join or combine or unite with others; increase the quality, quantity, size or scope of
1	1	2	115	State or say further
1	1	6	94	Constitute an addition
1	1	3	92	Bestow a quality on
2	2	4	47	Make an addition by combining numbers
2	2	5	44	Determine the sum of

Table 3: Senses for add-v in WN, OntoNotes (ON) and our clusters (Conf), including frequencies in train and glosses.

Conf	ON	WN	#	Gloss
1	2	1	449	a position on a scale of intensity or amount or quality
1	2	2	197	a relative position or degree of value in a graded group
1	6	3	143	a specific identifiable position in a continuum or series or especially in a process
5	5	7	18	an abstract place usually conceived as having depth
6	5	8	13	a structure consisting of a room or set of rooms at a single position along a vertical scale
2	1	4	11	height above ground
4	3	6	3	a flat surface at right angles to a plumb line
3	4	5	2	indicator that establishes the horizontal when a bubble is centered in a tube of liquid

Table 4: Senses for level-v in WN, OntoNotes (ON) and our clusters (Conf), including frequencies in train and glosses.

Conf	ON	WN	#	Gloss
1	1	1	285	give help or assistance; be of service
1	1	3	217	be of use
1	1	6	194	contribute to the furtherance of
1	1	2	80	improve the condition of
2	1	4	21	abstain from doing; always used with a negative
5	3	8	4	improve; change for the better
3	2	5	0	help to some food; help with food or drink
4	2	7	0	take or use

Table 5: Senses for help-n in WN, OntoNotes (ON) and our clusters (Conf), including frequencies in train and glosses.

lists those 18 words. We leave the analysis of the rest of words for further work, as they raise issues about overlapping sense boundaries, and our main goal is to check the quality of our method to group fine-grained senses.

Table 2 shows the statistics for those 18 words. Overall, the average polysemy of our clusters is higher and the performance of IMS on our clusters is also higher. We take this as an indication of the good quality of our clusters. On the other hand, the MFS on our clusters is considerably higher, which could mean that our algorithm has a tendency to lump together frequent senses, casting doubts on the quality of the clusters.

We selected three words for illustration, depending on the difference in number of clusters. Tables 3 to 5 show the senses of those four words[4]. In the case of add-v (Table 3), the clusters produced by our algorithm are the same as OntoNotes. For level-n (Table 4), although the number of clusters is the same, we group WordNet sense #3 together with senses #1 and #2, while OntoNotes keeps it separate. Note that sense #3 is very frequent, and as such it is lumped into a coarse grained sense which covers most of the occurrences. WordNet sense #8, on the contrary, is grouped by Ontonotes with #7, while we keep them separate. We think that in both cases, one could argue that our clusters make as much senses as those of OntoNotes, even if the distribution of our

[4]Note that coarse senses not in WordNet are not included.

cluster is more skewed than that of Ontonotes.

In the case of help-v (Table 5) our clusters produces more senses than those in Ontonotes. We think that sense #4, which is always used with a negative, can be sensibly considered a separate sense. Senses #5 and #7 are very similar, but being unattested in the train data, our algorithm is unable to cluster them.

In summary, the analysis of those (and other) examples shows that, in general, the sense clusters produced by our algorithm make sense. In a way, this was to be expected, as the clustering decisions depend on how often the volunteers confused the use of two senses. Our analysis also shows that our clustering does have an undesired tendency to cluster together frequent senses, while senses which occur rarely the train data are usually kept separate, adding artificially to the overall polysemy figure.

In the future we would like to study whether it is possible to make our algorithm more robust to this tendency to join frequent senses, perhaps discounting frequency from confusion measures.

7 Comparison to Supersenses

We also perform a qualitative study comparing our coarse grained senses to WN Supersenses. Supersenses are based on the lexicographer file names for WordNet, where all senses of the word that belong to the same lexicographer file (e.g. the artifact file) are joined together. They include 15 sense for verbs and 26 for nouns. Although WordNet also provide supersenses for adjective and adverbs, these are not semantically motivated and do not provide any higher abstraction (Johannsen et al., 2014).

Table 2 show the results for the target 45 words (adjectives included). The average polysemy of the supersenses is lower for all parts of speech with respect to our clustered senses and OntoNotes. Note that, word-wise, polysemy varies significantly: many words keep one or two senses, while others maintain high polysemy level (roughly similar to fine-grained senses). IMS and MFS performances are similar to OntoNotes.

Tables 6 to 8 show the differences in clustering for the same set of words (*add-v*, *level-n*, and *help-n*). In the case of add-v (Table 6), we produce two coarse grained sense against the 5 supersenses. The only coarse sense in Supersenses groups the arithmetic operation with *state or say further*, begin both *communication* senses, while our algorithm keeps groups them in separate sense clusters.

For level-n (Table 7) our algorithm produces more senses than the number of supersenses (6 *vs* 4). Supersenses of *state* and *attribute* are distributed between our clusters #1 and #2. Our clusters #3, #4 and #6 are lumped together as an *artifact*, although it would make sense to keep them separated. Finally, in the case of help-n (Table 8), we obtain the same amount of senses, but grouping differs considerably. For example, WordNet senses #3 and #4 are grouped under the *stative* supersense, although the definition and use of the two senses are completely different. On the other hand, our cluster #1 comprises the most frequent 4 senses.

Overall, the comparison of supersenses and our confusion-based coarse grained senses show complicated overlaps, contrary to OntoNotes, in which most of the clusters in one are subsumed in the other. Each of the sense groupings represent very different sense inventories. This shows the difficulty of having a *universal* sense representation that is useful for any application at hands. Actually, the choice of the inventory will depend on the angle of the meanings required by the application.

8 Conclusions and Future Work

This work explores the use of crowdsourced annotations to cluster senses that are often confused by non-expert annotators. Our method can provide clusters at different levels of granularity. We show that, for instance, we can construct clusters yielding around 90% accuracy for 45 words, with higher error reduction with respect to MFS than fine-grained senses. By construction, we merge senses which are often confused by annotators, yielding sensible sense clusters, as corroborated by manual inspection. The comparison to OntoNotes groupings fares well, with similar groupings, while the comparison to Supersenses shows that Supersenses follow a different grouping criterion, with overlapping clusters. The main weakness of our method seems to be the tendency to cluster together frequent senses.

This work is a small contribution towards the design of an ambitious annotation effort enabling

Conf	SS	WN	#	Gloss
1	change	1	421	make an addition (to); join or combine or unite with others; increase the quality, quantity, size or scope of
1	stative	6	94	constitute an addition
1	possession	3	92	bestow a quality on
1	communication	2	115	state or say further
2	communication	5	44	determine the sum of
2	cognition	4	47	make an addition by combining numbers

Table 6: Senses for add-v in WN, Supersenses (SS) and our clusters (Conf), including frequencies in train and glosses.

Conf	SS	WN	#	Gloss
1	state	2	197	a relative position or degree of value in a graded group
1	state	3	143	a specific identifiable position in a continuum or series or especially in a process
1	attribute	1	449	a position on a scale of intensity or amount or quality
2	attribute	4	11	height above ground
5	cognition	7	18	an abstract place usually conceived as having depth
6	artifact	8	13	a structure consisting of a room or set of rooms at a single position along a vert. scale
4	artifact	6	3	a flat surface at right angles to a plumb line
3	artifact	5	2	indicator that establishes the horizontal when a bubble is centered in a tube of liq.

Table 7: Senses for level-n in WN, Supersenses (SS) and our clusters (Conf), including frequencies in train and glosses.

Conf	SS	WN	#	Gloss
1	social	1	285	give help or assistance; be of service
1	social	6	194	contribute to the furtherance of
1	body	2	80	improve the condition of
1	stative	3	217	be of use
2	stative	4	21	abstain from doing; always used with a negative
5	change	8	4	improve; change for the better
3	consumption	5	0	help to some food; help with food or drink
4	consumption	7	0	take or use

Table 8: Senses for help-n in WN, Supersenses (SS) and our clusters (Conf), including freq. in train and glosses.

widespread use of high accuracy WSD. For the near future we would like to improve the error reduction with respect to the MFS trying to factor out sense frequency from clustering decisions. We would also like to check out-of-domain corpora, and to contrast the results of our confusion-based clusters with respect to other sense-clustering methods. Finally, we are aware that the final validity our technique needs to be shown in a downstream application.

Acknowledgments

This work was partially funded by MINECO (CHIST-ERA READERS project – PCIN-2013-002-C02-01, and SKaTeR project – TIN2012-38584-C06-02), and the European Commission (QTLEAP – FP7-ICT-2013.4.1-610516). The IXA group is funded by the Basque Government (A type Research Group).

References

Eneko Agirre and Oier Lopez de Lacalle. 2003. Clustering wordnet word senses. In *Proceedings of the Conference on Recent Advances on Natural Language Processing (RANLP'03)*, pages 11–18, Bulgary.

E. Agirre, T. Baldwin, and D. Martinez. 2008. Improving Parsing and PP attachment Performance with Sense Information. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL/HLT'08)*, pages 317–325, Columbus, USA.

A. P. Dawid and A. M. Skene. 1979. Maximum likelihood estimation of observer error-rates using the em algorithm. *Applied Statistics*, 28(1):20–28.

C. Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. Ontonotes: The 90% solution. In *Proceedings of HLT-NAACL 2006*, pages 57–60.

Anders Johannsen, Dirk Hovy, Héctor Martínez Alonso, Barbara Plank, and Anders Søgaard. 2014. More or less supervised supersense tagging of twitter. In *Proceedings of the Third Joint Conference on Lexical and Computational Semantics (*SEM 2014)*, pages 1–11, Dublin, Ireland, August. Association for Computational Linguistics and Dublin City University.

Yoong Keok Lee and Hwee Tou Ng. 2002. An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, pages 41–48. Association for Computational Linguistics, July.

Rada Mihalcea and Dan I. Moldovan. 1999. A method for word sense disambiguation of unrestricted text. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 152–158, College Park, Maryland, USA, June. Association for Computational Linguistics.

G.A. Miller, C. Leacock, R. Tengi, and R.Bunker. 1993. A Semantic Concordance. In *Proceedings of the workshop on Human Language Technology (HLT'93)*.

R. Navigli, K. C. Litkowski, and O. Hargraves. 2007. SemEval-2007 Task 07: Coarse-Grained English All-Words Task. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007) in conjunction with ACL*, pages 30–35, Prague, Czech Republic.

H. T. Ng and H. B. Lee. 1996. Integrating multiple knowledge sources to disambiguate word sense: an exemplar-based approach. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, ACL '96, pages 40–47, Stroudsburg, PA, USA. Association for Computational Linguistics.

Martha Palmer, Hoa Trang Dang, and Christiane Fellbaum. 2007. Making fine-grained and coarse-grained sense distinctions, both manually and automatically. *Natural Language Engineering*, 13(2):137–163.

Rebecca J. Passonneau and Bob Carpenter. 2014. The benefits of a model of annotation. *Transactions of the Association for Computational Linguistics*, 2(1-9):311–326.

Rebecca J. Passonneau, Collin F. Baker, Christiane Fellbaum, and Nancy Ide. 2012. The MASC word sense corpus. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Mehmet Uur Doan, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may. European Language Resources Association (ELRA).

W. Peters, I. Peters, , and P. Vossen. 1998. Automatic sense clustering in eurowordnet. In *Proceedings of the First International Conference on Language Resources and Evaluation*, pages 409–416, Granada, Spain.

Rion Snow, Sushant Prakash, Daniel Jurafsky, and Andrew Y. Ng. 2007. Learning to merge word senses. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 1005–1014, Prague, Czech Republic, June. Association for Computational Linguistics.

Noriko Tomuro. 2001. Tree-cut and a lexicon based on systematic polysemy. In *Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics*, Pittsburgh, USA.

Hwee Tou Ng, Chung Yong Lim, and Shou King Foo. 1999. Siglex99: Standardizing lexical resources.

Z. Zhong and H. T. Ng. 2010. It makes sense: A wide-coverage word sense disambiguation system for free text. In *Proceedings of the ACL 2010 System Demonstrations*, pages 78–83, Uppsala, Sweden.