Inconsistency Detection in Semantic Annotation

Nora Hollenstein, Nathan Schneider, Bonnie Webber
University of Edinburgh
School of Informatics
s1303835@ed.ac.uk, nschneid@inf.ed.ac.uk, bonnie@inf.ed.ac.uk

Abstract

Inconsistencies are part of any manually annotated corpus. Automatically finding these inconsistencies and correcting them (even manually) can increase the quality of the data. Past research has focused mainly on detecting inconsistency in syntactic annotation. This work explores new approaches to detecting inconsistency in *semantic* annotation. Two ranking methods are presented in this paper: a *discrepancy* ranking and an *entropy* ranking. Those methods are then tested and evaluated on multiple corpora annotated with multiword expressions and supersense labels. The results show considerable improvements in detecting inconsistency candidates over a random baseline. Possible applications of methods for inconsistency detection are improving the annotation procedure and guidelines, as well as correcting errors in completed annotations.

Keywords: inconsistency detection, annotation, error detection, corpora

1. Introduction

Inconsistencies can be found in any manually-annotated corpus due to underspecified or even missing guidelines, ambiguity, insufficient annotator expertise, and/or human errors. A consistent corpus is not only a high-quality lexical resource, but a foundation for robust automatic language processing via supervised learning. Since inconsistencies in the training corpus can lead to low performance, consistent annotation is of practical as well as theoretical benefit.

Annotation inconsistencies can be subclassified into annotation errors and hard cases, although they cannot always be easily distinguished (Plank et al., 2014). An **annotation error** is an instance that is annotated incorrectly according to the guidelines. Annotation errors have a correct answer. To give an example from part-of-speech annotation, consider the phrase *'a summer feeling'*. The word *'summer'* should be annotated as a common noun; marking it as a comparative adjective or a determiner would be an annotation error.

Linguistically hard cases are instances that do not have one correct answer; they may have multiple correct answers or it may not be obvious what the label should be. Such annotations are not necessarily incorrect, but can be ambiguous or underspecified in the guidelines. Beigman Klebanov and Beigman (2014) define *hard cases* as instances which are difficult to decide upon, which is when annotator preferences come into play, and as a consequence inter-annotator agreement and the consistency in the corpus are lower than in linguistically unambiguous cases.

Our concern here is the automatic detection of inconsistencies in semantic annotation. We consider datasets annotated for lexical semantic units (multiword expressions) and classes (supersenses); Section 2 gives an overview of these annotations. To rank lexical types that may be annotated inconsistently, we propose two methods, one based on absolute frequency and a second based on entropy (Section 3). Section 4 presents a precision-oriented evaluation of these methods on the lexical semantic datasets; Section 5 argues that these results indicate the viability of our methods for helping improve corpus consistency. We conclude with a discussion of related and future work.

2. Data sets

2.1. Multiword Expressions

Multiword expressions (MWEs) are expressions of at least two words whose combination is syntactically and/or semantically idiosyncratic in nature (Baldwin and Kim, 2010). Moreover, they act as a single unit at some level of linguistic analysis.

Encompassing 55,000 words of English web reviews, the **STREUSLE 2.0** corpus[1] has been comprehensively annotated for multiword expressions (Schneider et al., 2014). The annotation was done by six annotators, all of whom were linguists and native speakers of English. Every sentence was annotated by at least two annotators, with a gold standard created through joint annotator consensus. The average inter-annotator F_1 over all pairings of five annotators was 65%. A gold standard was created through joint annotator consensus.

The **Wiki50** corpus[2] consists of 50 Wikipedia articles in which several classes of multiword expressions and named entities are manually annotated (Vincze et al., 2011). The corpus contains 100,308 tokens and includes various topics.

Difficult cases that lead to annotation inconsistencies in these corpora are expressions such as prepositional verbs where it is not clear whether the preposition specifically belongs to the verb (e.g. *'listen to'*, *'look for'*) and nomi-

[1] http://www.cs.cmu.edu/ ark/LexSem/
[2] http://www.inf.u-szeged.hu/rgai/mwe

nal compounds that include more than two elements (e.g., *'pumpkin spice latte'* or *'surprise birthday party'*.

2.2. Supersense Labels

Supersenses are coarse-grained semantic classes such as PERSON, TIME, and ARTIFACT for nouns and MOTION, EMOTION, and COMMUNICATION for verbs (Ciaramita and Altun, 2006).[3] Supersense annotation is the task of assigning one of these labels to selected tokens in a corpus (Schneider et al., 2012).

In addition to MWEs, Schneider and Smith (2015) also annotated supersense labels for all verbs and nouns in the STREUSLE corpus, including all strong MWEs. As a second corpus, the publicly available Twitter data sets[4] by Johannsen et al. (2014) are annotated with supersenses using the BIO (Begin-Inside-Other) notation. In total this data set comprises 19,232 tokens.[5]

3. Methods

3.1. Scope and Definitions

We focus on types that have been labelled differently in different places in the corpus. We call these types "ambiguous". They may be truly ambiguous (i.e., polysemous), or the ambiguity may reflect inconsistency in their labelling. The present work seeks to automatically distinguish inconsistencies from linguistic ambiguities.

Multiword expressions. Formally, let $\mathcal{M}$ be the set of multiword expression types. Each $M \in \mathcal{M}$ is an ordered sequence of words (an n-gram, possibly with gaps) that is annotated as an MWE at least once in the corpus.
Character case is ignored when assigning sequences to types. For $M \in \mathcal{M}$, let n_M^+ be the number of times M is annotated as an MWE in the corpus, and n_M^- be the number of times the same sequence occurs in the corpus but is *not* annotated as an MWE.

Supersenses. Supersense tagging associates a coarse-grained semantic class (see Section 2.2) with a lexical expression, which may be a single word or MWE. There are multiple inventories of supersenses; in this work we focus on the 26 supersenses for nouns and the 15 for verbs.

Let $\mathcal{S}$ be the set of expressions that are supersense-tagged anywhere in the corpus. Let n_S^ℓ denote the number of times the expression $S \in \mathcal{S}$ is annotated with supersense label ℓ. Define $n_S = \sum_\ell n_S^\ell$, the total number of supersense-labeled tokens of S.

[3] Supersense inventories have also been proposed for adjectives and prepositions (Tsvetkov et al., 2014; Schneider et al., 2015), though supersense annotations are limited to nouns and verbs in the data sets we use.

[4] https://github.com/coastalcph/supersense-data-twitter

[5] A new, larger corpus that includes STREUSLE was compiled for the DiMSUM 2016 shared task on MWE identification and supersense tagging (Schneider et al., 2016): `https://github.com/dimsum16/dimsum-data`

Both methods described below for detecting potential inconsistencies are based on ranking each ambiguous type in terms of differences in how its tokens have been labelled.

3.2. Weighted Discrepancy Ranking

Discrepancy is expressed as the difference between the proportion of favorable evidence (e.g., the number of times an expression M occurs annotated as an MWE) minus the proportion of unfavorable evidence (e.g., the number of times M occurs not annotated as an MWE).

3.2.1. Weighted Discrepancy Ranking for MWE Annotations

As already noted, $M \in \mathcal{M}$ is the set of tokens that have been annotated at least once in the STREUSLE corpus for a given sequence of words. For the sequence *'in fact'* (from the STREUSLE corpus), $\mathcal{M}$ includes the following tokens:

- I saw deer frequently, **in_fact** a small herd were grazing ...
- **In_fact** I look_forward_to taking my animals to the vet ...
- ... when **in fact** they are a very average place at_best ...

Note that the underscore indicates an MWE annotation; the third of these examples is not annotated as an MWE.

In the STREUSLE corpus 10.8% of all annotated multiword expression types occur not annotated at least once, while the Wiki50 corpus contains only 1.2% of such types.

For each type $M \in \mathcal{M}$, its **weighted discrepancy** score is computed as

$$wdiscrep(M) = |n_M^+ - n_M^-| \cdot n_M^+ \quad (1)$$

For instance, the expression *'a couple'* occurs 15 times, 13 times annotated as an MWE. It receives a resulting score of *wdiscrep*(M) = 143.

The main assumption for the above equation is that a high discrepancy between frequencies of annotated and not annotated occurrences is an indication that the type is inconsistently annotated. This discrepancy is weighted (scaled) by the number of times the MWE was annotated to put more weight on frequent multiword expressions.

With the above measure, we rank types in descending order. The hypothesis is that types with greater annotated-MWE frequency overall, and greater discrepancy between annotated and unannotated tokens, are more likely to contain inconsistently labeled tokens.

3.2.2. Weighted Discrepancy Ranking for Supersense Annotations

A similar weighted discrepancy ranking method can be defined for supersense labels.

A type $S \in \mathcal{S}$ is defined as a word form—noun or verb—paired with a single POS tag. For instance, in the Twitter corpus, the type $\langle$*computer*, NN$\rangle$ (NN = singular noun) includes the following tokens:

- I can go a week w/o tv, phones or a **computer**. (NOUN.COMMUNICATION)

- I have this on my **computer**. (NOUN.ARTIFACT)

In the STREUSLE corpus, 12.82% of all annotated noun and verb types have more than one supersense label, while the Twitter corpus contains 5.12%.

In defining a weighted discrepancy measure for supersense tags, we consider both the most and least common labels and the number of distinct labels, in order to better separate linguistically ambiguous types from types with annotation inconsistencies. The denominator counts the number of distinct labels seen with S. We only score types occurring with at least two distinct supersense labels.

$$\textit{wdiscrep}(S) = \frac{(\max_\ell n_S^\ell) - (\min_\ell n_S^\ell \geq 1)}{|\{\ell \mid n_S^\ell > 0\}|} \cdot n_S \quad (2)$$

We compute this separately for nouns and verbs, and in each case rank types in descending order on the hypothesis that a higher score will indicate a greater likelihood of inconsistency. The noun *'job'*, for example, occurs 50 times with three different supersense labels, which results in a score of *wdiscrep*(S) = 700.

3.3. Entropy Ranking

Entropy is an information theoretic measure of uncertainty (Jurafsky and Martin, 2000). Here we use it as a basis for a second ranking method for inconsistency detection. We use the above notation for supersenses, though the same formulation works for multiword expressions where + and − are the labels.

For a given type S, entropy is defined as

$$H_S = -\sum_\ell p_S(\ell) \log_2 p_S(\ell) \quad (3)$$

where the probability that a type S is labeled with ℓ is estimated by relative frequency:

$$p_S(\ell) = \frac{n_S^\ell}{n_S} \quad (4)$$

Because we consider only types with at least two distinct labels in the corpus, no label will have a probability of 1, and the entropy over labels will always be positive.

For instance, the expression *'have been'* occurs once annotated as an MWE, and 50 times without annotations. Hence, the resulting entropy H_S = 0.14. We might conclude from its low entropy that it would be worth checking whether the single annotated token of *'have been'* is a rare case or an error. If the latter, correcting it will increase annotation consistency for this type.

High entropy corresponds to high uncertainty as to the label. We rank types in *ascending* order, hypothesizing that small (but still nonzero) entropy indicates a highly skewed distribution over labels, and therefore a greater likelihood of anomalous (inconsistent) annotations.

4. Results

For each corpus and ranking method, we reviewed the annotation of the top-ranking 50 types. For each type, all sentences containing at least one token of that type were examined to determine if the tokens were annotated consistently. If any errors or inconsistently annotated "hard cases" were found in any of these sentences, the type was marked as a *true inconsistency*. One ranking is better than another if true inconsistencies appear higher in the ranking. We measure this by calculating precision at different ranks (i.e., precision@k) and compare it to a baseline of randomly ranked types.

Results appear in Tables 1–2 for each data set and annotation type. Quantitatively, we find that both the discrepancy ranking and the entropy ranking perform better than randomly ranking types. Moreover, the precision for the discrepancy ranking is slightly higher than the entropy ranking in all cases except for MWEs in Wiki50, where the two methods are essentially the same.

A large number of MWE types exhibiting true inconsistencies are ranked within the top 50. Many are verbal constructions. For instance, types consisting of a verb and a preposition (e.g., *'work with'*, *'come in'*, *'leave for'*, *'look for'*) and light verb constructions (*'make sure'*, *'take care'*) are prone to annotation inconsistencies. Types composed of a determiner followed by a quantifier (*'a few'*, *'a little'*, *'a couple'*, *'a bit'*) also contain many inconsistencies.

In the supersense annotations, most inconsistencies arise between the following labels: LOCATION vs. GROUP (e.g., *'restaurant'*, *'place'*), TIME vs. EVENT (*'birthday'*, *'night'*) and PERSON vs. GROUP (*'staff'*, *'followers'*).

Errors. Both inconsistency detection methods for MWEs returned expressions such as *'how to'*, *'to go'* and *'kind of'*, but these are false positives. For instance, in the sentence *'I have to go now.'*, *'to go'* is not an MWE, while it has been correctly annotated as an MWE in the sentence *'I'd like my coffee to go.'*.
False positives from both ranking methods for the supersense annotations included linguistically ambiguous words such as *'end'*, *'place'* or *'stuff'*. The word *'stuff'* has different meanings depending on the context. Therefore, it can be labeled correctly with multiple supersenses. For example, in the sentence *'I have stuff to do.'* it is annotated as ACT, while in *'I've got the stuff to make egg rolls.'* it is labeled as SUBSTANCE.

5. Discussion

The results presented in the previous section indicate that ranking methods are successful in outputting a large number of inconsistency candidates. Hence, the consistency of a corpus can be improved with a reasonable amount of effort: a large number of inconsistencies can be retrieved and corrected without having to sift through too many types that are truly linguistically ambiguous. Our proposed ranking methods achieve higher precision

k	baseline	discrepancy	entropy
10	0.80	1.00	0.70
20	0.75	1.00	0.85
30	0.77	0.99	0.90
40	0.75	0.93	0.90
50	0.78	0.92	0.90

(a) STREUSLE

k	baseline	discrepancy	entropy
10	0.90	0.90	1.00
20	0.90	0.95	0.95
30	0.90	0.93	0.93
40	0.88	0.95	0.95
50	0.90	0.92	0.92

(b) Wiki50

Table 1: Multiword expression results. Fraction of returned MWEs found to be inconsistent at each rank (k=10, 20, etc.), when ranking randomly (baseline), by discrepancy and by entropy. Figures given for STREUSLE (where 10.8% of annotated MWE types are ambiguous, 275 types), and Wiki50 (where ambiguity by type is only 1.2%, 66 types).

k	baseline	discrepancy	entropy
10	0.50	0.70	0.40
20	0.35	0.50	0.50
30	0.33	0.60	0.53
40	0.38	0.65	0.60
50	0.42	0.72	0.60

(a) STREUSLE

k	baseline	discrepancy	entropy
10	0.80	0.90	0.80
20	0.80	0.85	0.75
30	0.77	0.80	0.77
40	0.73	0.80	0.78
50	0.72	0.76	0.76

(b) Twitter

Table 2: Noun supersense results. Fraction of returned noun supersenses found to be inconsistent at each rank (k=10, 20, etc.), when ranking randomly (baseline), by discrepancy and by entropy. Figures given for STREUSLE (where 18.6% of annotated MWE types are ambiguous, 881 types), and Twitter corpus (where ambiguity by type is only 6.9%, 216 types).

for MWE annotations than for supersense annotations. This might be due to the fact that MWE labels are binary, while multiple supersense labels can be assigned to the same word. Presumably, inconsistencies are more likely to appear in multi-label annotation tasks and may be harder to detect because of higher ambiguity.

The proposed ranking methods for inconsistency detection in semantic annotations can be applied in different scenarios. First, they can be applied to enhance annotation tools. Annotation procedures could be greatly improved if the annotators were shown all other instances of the same type that have already been annotated, warning of a possible inconsistency. Thus it would be possible to increase consistency during the annotation process and not only on completely annotated corpora. In such a scenario, methods for identifying inconsistent types or tokens could save valuable adjudication and revision time by working prophylactically.

Furthermore, the insights of the ranking and evaluation process of this work can be used to improve the annotation guidelines. Through systematically analyzing the corpora it becomes easier to clarify and expand the guidelines and to include real examples that may have been overlooked previously. And finally, through manual revision of the highest ranked types, it will be possible to increase corpus consistency by targeting the most frequent types and revising all instances of a type at the same time.

6. Prior Work

Finding inconsistencies requires identifying similar instances that have been labeled differently. A prominent family of methods considers **variation n-grams**—word sequences that receive different labelings in different parts of the corpus—and heuristically rank them to identify likely errors (Dickinson and Meurers, 2003; Boyd et al., 2007). Here, we consider ambiguous types, which in the case of supersenses are variation 1-grams. (For MWEs the analogy doesn't hold as well because an n-gram of two or more words receives a label as a unit.)

Most previous work on error detection in corpus annotation has focused on syntactic annotations—POS tags (Loftsson, 2009; Eskin, 2000; Ma et al., 2001) and parses (Ule and Simov, 2004; Kato and Matsubara, 2010)—rather than semantic annotations. Dickinson and Lee (2008) is one exception: the authors considered inconsistencies in annotations of predicate-argument structures. To our knowledge, no previous methods have been developed for inconsistencies in lexical semantic segmentation or tagging.

Previous work has found benefit to considering *context* when determining whether an ambiguous expression is inconsistently annotated (Nakagawa and Matsumoto, 2002). For instance, Nguyen et al. (2015) applied an entropy-based scoring method that is similar to our entropy measure (3.3), except it conditions on contextual features. When it comes to lexical semantic annotation, we leave to future work the possibility of exploiting context to detect inconsistencies, though the benefits of doing so may be limited for our small corpora.

Other approaches have taken advantage of *multiple annotations* from different annotators (Hovy et al., 2013; Passonneau and Carpenter, 2014). Our methods only consider one annotation per sentence, and therefore do not depend on information which is available only for some corpora.

7. Conclusion

Since inconsistency detection in semantic annotation is a largely unexplored topic and semantic inconsistencies are more difficult to grasp than syntactic inconsistencies,

we explore two ranking-based methods to approach this task. We apply both methods to annotations of multiword expressions and supersense labels on different data sets. Overall, the proposed ranking methods are successful in detecting inconsistency candidates with high precision. Furthermore, this simple approach does not require extensive preprocessing.

The data sets used in this project provided annotations of multiword expressions and supersense labels. The presented corpora were used to test and evaluate the ranking methods. We aggregated the annotations into types, which were ranked in their likelihood of containing annotation inconsistencies. We manually evaluated how many of the highest-ranking types actually contain annotation inconsistencies. Additional results and analyses can be found in Hollenstein (2015) and the data is available online[6].

The current work takes a first step towards ensuring that semantic annotation is consistent. The next and harder stop involves automating the identification of inconsistent tokens. This should be addressed in future research. We believe that our ranking approach is general and can be applied to other forms of semantic annotation. Future research should thus also explore its generalizability (e.g. for syntactic annotation).

8. Bibliographical References

Baldwin, T. and Kim, S. N. (2010). Multiword expressions. In Nitin Indurkhya et al., editors, *Handbook of Natural Language Processing, Second Edition*, pages 267–292. CRC Press, Taylor and Francis Group, Boca Raton, FL.

Beigman Klebanov, B. and Beigman, E. (2014). Difficult cases: from data to learning, and back. In *Proc. of ACL*, pages 390–396.

Boyd, A., Dickinson, M., and Meurers, D. (2007). Increasing the recall of corpus annotation error detection. In *Proc. of TLT*, volume 1, pages 19–30.

Ciaramita, M. and Altun, Y. (2006). Broad-coverage sense disambiguation and information extraction with a supersense sequence tagger. In *Proc. of EMNLP*, pages 594–602.

Dickinson, M. and Lee, C. M. (2008). Detecting errors in semantic annotation. In *Proc. of LREC*, pages 605–610.

Dickinson, M. and Meurers, W. D. (2003). Detecting inconsistencies in treebanks. In *Proc. of TLT*, volume 3, pages 45–56.

Eskin, E. (2000). Detecting errors within a corpus using anomaly detection. In *Proc. of NAACL*, pages 148–153.

Hollenstein, N. (2015). *Inconsistency Detection in Semantic Annotation*. MSc. dissertation, University of Edinburgh, Edinburgh, UK, August. http://project-archive.inf.ed.ac.uk/msc/20150152/msc_proj.pdf.

Hovy, D., Berg-Kirkpatrick, T., Vaswani, A., and Hovy, E. (2013). Learning whom to trust with MACE. In *Proc. of NAACL-HLT*, pages 1120–1130.

Johannsen, A., Hovy, D., Martínez Alonso, H., Plank, B., and Søgaard, A. (2014). More or less supervised supersense tagging of Twitter. In *Proc. of *SEM*, pages 1–11.

Jurafsky, D. and Martin, J. H. (2000). *Speech and Language Processing*. Pearson, international edition.

Kato, Y. and Matsubara, S. (2010). Correcting errors in a treebank based on synchronous tree substitution grammar. In *Proc. of ACL*, pages 74–79.

Loftsson, H. (2009). Correcting a POS-tagged corpus using three complementary methods. In *Proc. of EACL*, pages 523–531.

Ma, Q., Lu, B., Murata, M., Ichikawa, M., and Isahara, H. (2001). On-line error detection of annotated corpus using modular neural networks. In *Artificial Neural Networks — ICANN 2001*, pages 1185–1192.

Nakagawa, T. and Matsumoto, Y. (2002). Detecting errors in corpora using support vector machines. In *Proc. of COLING*, pages 1–7.

Nguyen, P. T., Le, A. C., Ho, T. B., and Nguyen, V. H. (2015). Vietnamese treebank construction and entropy-based error detection. *Language Resources and Evaluation*, 49(3):487–519.

Passonneau, R. J. and Carpenter, B. (2014). The benefits of a model of annotation. *Transactions of the Association for Computational Linguistics*, 2:311–326, October.

Plank, B., Hovy, D., and Søgaard, A. (2014). Linguistically debatable or just plain wrong? In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, volume 2, pages 507–511.

Schneider, N. and Smith, N. A. (2015). A corpus and model integrating mutiword expressions and supersenses. In *Proc. of NAACL-HLT*.

Schneider, N., Mohit, B., Oflazer, K., and Smith, N. A. (2012). Coarse lexical semantic annotation with supersenses: an Arabic case study. In *Proc. of ACL*, pages 253–258.

Schneider, N., Onuffer, S., Kazour, N., Danchik, E., Mordowanec, M. T., Conrad, H., and Smith, N. A. (2014). Comprehensive annotation of multiword expressions in a social web corpus. In *Proc. of LREC*, pages 455–461.

Schneider, N., Srikumar, V., Hwang, J. D., and Palmer, M. (2015). A hierarchy with, of, and for preposition supersenses. In *Proc. of The 9th Linguistic Annotation Workshop*, pages 112–123.

Schneider, N., Hovy, D., Johannsen, A., and Carpuat, M. (2016). SemEval 2016 Task 10: Detecting Minimal Semantic Units and their Meanings (DiMSUM). In *Proc. of SemEval*.

Tsvetkov, Y., Schneider, N., Hovy, D., Bhatia, A., Faruqui, M., and Dyer, C. (2014). Augmenting English adjective senses with supersenses. In Nicoletta Calzolari, et al., editors, *Proc. of LREC*, pages 4359–4365.

Ule, T. and Simov, K. (2004). Unexpected productions may well be errors. In *Proc. of LREC*, pages 1795–1798.

Vincze, V., Nagy T., I., and Berend, G. (2011). Multiword expressions and named entities in the Wiki50 corpus. In *Proc. of RANLP*, pages 289–295.

[6] https://github.com/norahollenstein/inconsistency-detection