Chapter 9
Semi-automated resolution of inconsistency for a harmonized multiword-expression and dependency-parse annotation

Julian Brooke
The University of Melbourne

King Chan
The University of Melbourne

Timothy Baldwin
The University of Melbourne

This chapter presents a methodology for identifying and resolving various kinds of inconsistency in the context of merging dependency and multiword expression (MWE) annotations, to generate a dependency treebank with comprehensive MWE annotations. Candidates for correction are identified using a variety of heuristics, including an entirely novel one which identifies violations of MWE constituency in the dependency tree, and resolved by arbitration with minimal human intervention. Using this technique, we identified and corrected several hundred inconsistencies across both parse and MWE annotations, representing changes to a significant percentage (well over 10%) of the MWE instances in the joint corpus and a large difference in MWE tagging performance relative to earlier versions.

1 Introduction

The availability of gold-standard annotations is important for the training and evaluation of a wide variety of Natural Language Processing (NLP) tasks, includ-
ing the evaluation of dependency parsers (Buchholz & Marsi 2006). In recent years, there has been a focus on multi-annotation of a single corpus, such as joint syntactic, semantic role, named entity, coreference and word sense annotation in Ontonotes (Hovy et al. 2006) or constituency, semantic role, discourse, opinion, temporal, event and coreference (among others) annotation of the Manually Annotated Sub-Corpus of the American National Corpus (Ide et al. 2010). As part of this, there has been an increased focus on harmonizing and merging existing annotated data sets as a means of extending the scope of reference corpora (Ide & Suderman 2007; Declerck 2008; Simi et al. 2015). This effort sometimes presents an opportunity to address conflicts among annotations, a worthwhile endeavour since even a small number of errors in a gold-standard syntactic annotation can, for example, result in significant changes in downstream applications (Habash et al. 2007). This chapter presents the results of a harmonization effort for the overlapping STREUSLE annotation (Schneider, Onuffer, et al. 2014) of multiword expressions (MWEs: Baldwin & Kim 2010) and dependency parse structure in the English Web Treebank (EWT: Bies et al. 2012), with the long-term goal of building reliable resources for joint MWE/syntactic parsing (Constant & Nivre 2016).

As part of merging these two sets of annotations, we use analysis of cross-annotation and type-level consistency to identify instances of potential annotation inconsistency, with an eye to improving the quality of the component and combined annotations. It is important to point out that our approach to identifying and handling inconsistencies does not involve re-annotating the corpus; instead we act as arbitrators, resolving inconsistency in only those cases where human intervention is necessary. Our three methods for identifying potentially problematic annotations are:

• a cross-annotation heuristic that identifies MWE tokens whose parse structure is incompatible with the syntactic annotation of the MWE;
• a cross-type heuristic that identifies n-grams with inconsistent token-level MWE annotations; and
• a cross-type, cross-annotation heuristic that identifies MWE types whose parse structure is inconsistent across its token occurrences.

The first of these is specific to this harmonization process, and as far as we are aware, entirely novel. The other two are adaptions of an approach to improving syntactic annotations proposed by Dickinson & Meurers (2003). After applying these heuristics and reviewing the candidates, we identified hundreds of errors in MWE annotation and about a hundred errors in the original syntactic annotations. We make available a tool that applies these fixes in the process of joining the two annotations into a single harmonized, corrected annotation, and release
9 Semi-automated resolution of annotation inconsistency

the harmonized annotations in the form of HAMSTER (the HArmonized Multiword and Syntactic Tree Resource): https://github.com/eltimster/HAMSTER. This chapter goes beyond the MWE2017 paper that first introduced HAMSTER (Chan et al. 2017) to show that the application of these and other corpus fixes has a major effect on MWE identification performance: we find that almost a quarter of the error originally assumed to be tagger error is actually attributable to errors in the corpus.

2 Related work

Our long-term goal is building reliable resources for joint MWE/syntactic parsing. Explicit modelling of MWEs has been shown to improve parser accuracy (Nivre 2004; Seretan & Wehrli 2006; Finkel & Manning 2009; Korkontzelos & Manandhar 2010; Green et al. 2013; Vincze et al. 2013; Wehrli 2014; Candito & Constant 2014; Constant & Nivre 2016). Treatment of MWEs has typically involved parsing MWEs as single lexical units (Nivre 2004; Eryiğit et al. 2011; Fotopoulou et al. 2014), but this flattened, “words with spaces” (Sag et al. 2002) approach is inflexible in its coverage of MWEs where components have some level of flexibility.

The English Web Treebank (Bies et al. 2012) represents a gold-standard annotation effort over informal web text. The original syntactic constituency annotation of the corpus was based on hand-correcting the output of the Stanford Parser (Manning et al. 2014); for our purposes we have converted this into a dependency parse using the Stanford Typed Dependency converter (de Marneffe et al. 2006). We considered the use of the Universal Dependencies representation (Nivre et al. 2016), but we noted that several aspects of that annotation (in particular the treatment of all prepositions as case markers dependent on their noun) make it inappropriate for joint MWE/syntactic parsing since it results in large numbers of MWEs that are non-continuous in their syntactic structure (despite being continuous at the token level).1 As such, the Stanford Typed Dependencies is the representation which has the greatest currency for joint MWE/syntactic parsing work (Constant & Nivre 2016).

The STREUSLE corpus (Schneider, Onuffer, et al. 2014) is based entirely on the Reviews subset of the EWT, and comprises 3,812 sentences representing 55,579 tokens. The annotation was completed by six linguists who are native English

---

1 An example of this would be a phrase such as think of home, where we consider think of to be an MWE, but the Universal Dependencies framework would treat of as a syntactic dependent of home, not think.
Deep tissue helps with pain in neck and shoulders.

Figure 1: An example where the arc count heuristic is breached. Deep tissue has been labeled in the sentence here as an MWE in STREUSLE. Deep and tissue act as modifiers to massage, a term that has not been included as part of the MWE.

speakers. Every sentence was assessed by at least two annotators, which resulted in an average inter-annotator F1 agreement of 0.7. The idiosyncratic nature of MWEs lends itself to challenges associated with their interpretation, and this was readily acknowledged by those involved in the development of the STREUSLE corpus (Hollenstein et al. 2016). Two important aspects of the MWE annotation are that it includes both continuous and non-continuous MWEs (e.g., check * out), and that it supports both weak and strong annotation. With regards to the latter, a variety of cues are employed to determine associative strength. The primary factor relates to the degree in which the expression is semantically opaque and/or morphosyntactically idiosyncratic. An example of a strong MWE would be top notch, as used in the sentence: We stayed at a top notch hotel. The semantics of this expression are not immediately predictable from the meanings of top and notch. On the other hand, the expression highly recommend is considered to be a weak expression as it is largely compositional – one can highly recommend a product – as indicated by the presence of alternatives such as greatly recommend which are also acceptable though less idiomatic. A total of 3,626 MWE instances were identified in STREUSLE, across 2,334 MWE types.

Other MWE-aware dependency treebanks include the various UD treebanks (Nivre et al. 2016), the Prague Dependency Treebank (Bejček et al. 2013), the Redwoods Treebank (Oepen et al. 2002), and others (Nivre 2004; Eryiğit et al. 2011; Candito & Constant 2014). The representation of MWEs, and the scope of types covered by these treebanks, can vary significantly. For example, the internal syntactic structure may be flattened (Nivre 2004), or in the case of Candito & Constant (2014), allow for distinctions in the granularity of syntactic representation for regular vs. irregular MWE types.

The identification of inconsistencies in annotation requires comparisons to be made between similar instances that are labeled differently. Boyd et al. (2007) employed an alignment-based approach to assess differences in the annotation
of \( n \)-gram word sequences in order to establish the likelihood of error occurrence. Other work in the syntactic inconsistency detection domain includes those related to POS tagging (Loftsson 2009; Eskin 2000; Ma et al. 2001) and parse structure (Ule & Simov 2004; Kato & Matsubara 2010). Dickinson & Meurers (2003) outline various approaches for detecting inconsistencies in parse structure within treebanks.

In general, inconsistencies associated with MWE annotation fall under two categories: (1) annotator error (i.e. false positives and false negatives); and (2) ambiguity associated with the assessment of hard cases. While annotation errors apply to situations where a correct label can be applied but is not, hard cases are those where the correct label is inherently difficult to assign. We address both these categories in this work.

3 Error candidate identification

3.1 MWE syntactic constituency conflicts

The hypothesis that drives our first analysis is that for nearly all MWE types, the component words of the MWE should be syntactically connected, which is to say that every word is a dependent of another word in the MWE, except one word which connects the MWE to the rest of the sentence (or the root of the sentence). We can realise this intuition by using an arc-count heuristic: for each labeled MWE instance we count the number of incoming dependency arcs that are headed by a term outside the MWE, and if the count is greater than one, we flag it for manual analysis. Figure 1 gives an example where the arc count heuristic is breached since both terms of the MWE *deep tissue* act as modifiers to the head noun that sits outside the MWE.

3.2 MWE type inconsistency

Our second analysis involves first collecting a list of all MWE types in the STREUSLE corpus, corresponding to lemmatized \( n \)-grams, possibly with gaps. We then match these \( n \)-grams across the same corpus, and flag any MWE type which has at least one inconsistency with regards to the annotation. That is, we extract as candidates any MWE types where there were at least two occurrences of the corresponding \( n \)-gram in the corpus that were incompatible with respect to their annotation in STREUSLE, including discrepancies in weak/strongish designation. For non-continuous MWE types, matches containing up to 4 words of intervening context between the two parts of the MWE type were included as candidates.
for further assessment. Some examples of many $n$-gram types which showed inconsistency in their MWE annotation include $\text{interested in, high quality, ask for, in town, pizza place, even though}$, and $\text{easy to work with}$.

### 3.3 MWE type parse inconsistency

The hypothesis that drives our third analysis is that we would generally expect the internal syntax of an MWE type to be consistent across all its instances. For each MWE type, we extracted the internal dependency structure of all its labeled instances, and flagged for further assessment any type for which the parse structure (including typed dependency label) varied between at least two of those instances. Note that although this analysis is aimed at fixing parse errors, it makes direct use of the MWE annotation provided by STREUSLE to greatly limit the scope of error candidates to those which are most relevant to our interest. Some MWE types which showed syntactic inconsistency include $\text{years old, up front, set up, check out, other than}$, and $\text{get in touch with}$.

### 4 Error arbitration

Error arbitration was carried out by the authors (all native English speakers with experience in MWE identification), with at least two authors looking at each error candidate in most instances, and for certain difficult cases, the final annotation being based on discussion among all three authors. One advantage of our arbitration approach over a traditional token-based annotation was that we could enforce consistency across similar error candidates (e.g., $\text{disappointed with}$ and $\text{happy with}$) and also investigate non-candidates to arrive at a consensus; where at all possible, our changes relied on precedents that already existed in the relevant annotation.

Arbitration for the MWE syntax conflicts usually involved identifying an error in one of the two annotations, and in most cases this was relatively obvious. For instance, in the candidate $\text{... the usual lady called in sick hours earlier, called in sick}$ was correctly labeled as an MWE, but the parse incorrectly includes $\text{sick}$ as a dependent of $\text{hours}$, rather than $\text{called in}$. An example of the opposite case is $\text{... just to make the appointment ...}$, where $\text{make the}$ had been labeled as an MWE, an obvious error which was caught by our arc count heuristic. There were cases where our arc count heuristic was breached due to what we would view as a

---

*2*Noting that we would not expect this to occur between MWE instances of a given combination of words, and non-MWE combinations of those same words.
general inadequacy in the syntactic annotation, but we decided not to effect a change because the impact would be too far reaching; examples of this were certain discourse markers (e.g., as soon as), and infinitives (e.g., have to complete where the to is considered a dependent of its verb rather than of the other term in the MWE have to). The most interesting cases were a handful of non-continuous MWEs where there was truly a discontinuity in the syntax between the two parts of the MWE, for instance no amount of can. This suggests a basic limitation in our heuristic, although the vast majority of MWEs did satisfy it.

For the two type-level arbitrations, there were cases of inconsistency upheld by real usage differences (e.g., a little house vs. a little tired). We identified clear differences in usage first and divided the MWE types into sets, excluding from further analysis non-MWE usages of MWE type n-grams. For each consistent usage of an MWE type, the default position was to prefer the majority annotation across the set of instances, except when there were other candidates that were essentially equivalent: for instance, if we had relied on majority annotation for job* do (e.g., the job that he did) it would have been a different annotation than do* job (e.g., do a good job), so we considered these two together. We treated continuous and non-continuous versions of the same MWE type in the same manner.

In the MWE type consistency arbitration, for cases where majority rules did not provide a clear answer and there was no overwhelming evidence for non-compositionality, we introduced a special internal label called hard. These correspond to cases where the usage is consistent and the inconsistency seems to be a result of the difficulty of the annotation item (as discussed earlier in Section 2), which extended also to our arbitration. Rather than enforce a specific annotation without strong evidence or allow the inconsistency to remain when there is no usage justification for it, the corpus merging and correction tool gives the user the option to treat hard annotated MWEs in varying ways: the annotation may be kept unchanged, removed, converted to weak, or converted to hard for the purpose of excluding it from evaluation. Examples of hard cases include go back, go in, more than, talk to, speak to, thanks guys, not that great, pleased with, have* option, get* answer, fix* problem. On a per capita basis, inconsistencies are more common for non-continuous MWEs relative to their continuous counterparts, and we suspect that this is partially due to their tendency to be weaker, in addition to the challenges involved in correctly discerning the non-continuous parts, which are sometimes at a significant distance from each other.

Table 1 provides a summary of changes to MWE annotation at the MWE type and token levels. Mixed refer to MWEs that are heterogeneous in the associative
strength between terms in the MWE (between weak and strongish). Most of the changes in Table 1 (98% of the types) were the result of our type consistency analysis. Almost half of the changes involved the use of the hard label, but even excluding these (since only some of these annotations required actual changes in the final version of the corpus) our changes involve over 10% of the MWE tokens in the corpus, and thus represent a significant improvement to the STREUSLE annotation.

Relative to the changes to the MWE annotation, the changes to the parse annotation were more modest, but still not insignificant: for 161 MWE tokens across 72 types, we identified and corrected a dependency and/or POS annotation error. The majority of these (67%) were identified using the arc count heuristic. Note we applied the parse relevant heuristics after we fixed the MWE type consistency errors, ensuring that MWE annotations that were added were duly considered for parse errors.

Table 1: Summary of changes to MWE annotation at the MWE type and token level.

<table>
<thead>
<tr>
<th></th>
<th>No MWE</th>
<th>Weak</th>
<th>Strong</th>
<th>Mixed</th>
<th>Hard</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Token</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>No MWE</td>
<td>—</td>
<td>55</td>
<td>136</td>
<td>6</td>
<td>151</td>
<td>348</td>
</tr>
<tr>
<td>Weak</td>
<td>35</td>
<td>—</td>
<td>22</td>
<td>4</td>
<td>46</td>
<td>107</td>
</tr>
<tr>
<td>Strong</td>
<td>44</td>
<td>42</td>
<td>—</td>
<td>9</td>
<td>70</td>
<td>165</td>
</tr>
<tr>
<td>Mixed</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>12</td>
<td>2</td>
<td>23</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td>81</td>
<td>101</td>
<td>161</td>
<td>31</td>
<td>269</td>
<td>643</td>
</tr>
<tr>
<td><strong>Type</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>No MWE</td>
<td>—</td>
<td>31</td>
<td>74</td>
<td>5</td>
<td>64</td>
<td>174</td>
</tr>
<tr>
<td>Weak</td>
<td>31</td>
<td>—</td>
<td>13</td>
<td>4</td>
<td>35</td>
<td>83</td>
</tr>
<tr>
<td>Strong</td>
<td>34</td>
<td>28</td>
<td>—</td>
<td>7</td>
<td>43</td>
<td>112</td>
</tr>
<tr>
<td>Mixed</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>7</td>
<td>2</td>
<td>18</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td>67</td>
<td>63</td>
<td>90</td>
<td>23</td>
<td>144</td>
<td>387</td>
</tr>
</tbody>
</table>

5 Experiments

In this section we investigate the effect of the HAMSTER MWE inconsistency fixes on the task of MWE identification. For this we use the AMALGr MWE identification tool of Schneider, Danchik, et al. (2014), which was developed on
the initial release of the STREUSLE (called then the CMWE).\textsuperscript{3} AMALGr is a supervised structured perceptron model which makes use of external resources including 10 MWE lexicons as well as Brown cluster information. For all our experiments we use the default settings from Schneider, Danchik, et al. (2014), including the original train/test splits and automatic part-of-speech tagging provided by the ARK TweetNLP POS tagger (Owoputi et al. 2013) trained on the all non-review sections of the English Web Treebank. We note that in contrast to typical experiments in NLP, here we are holding the approach constant while varying the quality of the dataset, which provides a quantification of the extent to which errors in the dataset interfered with our ability to build or accurately evaluate models. Following Schneider, Danchik, et al. (2014), we report an F-score which is calculated based on links between words: a true positive occurs when two words which are supposed to appear together in an MWE do so as expected.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>F1-score (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMWE (Schneider, Danchik, et al. 2014)</td>
<td>59.4</td>
</tr>
<tr>
<td>STREUSLE 3.0</td>
<td>64.6</td>
</tr>
<tr>
<td>HAMSTER-original</td>
<td>69.1</td>
</tr>
<tr>
<td>HAMSTER-notMWE</td>
<td>68.2</td>
</tr>
<tr>
<td>HAMSTER-weak</td>
<td>69.4</td>
</tr>
<tr>
<td>HAMSTER-original-noeval</td>
<td>70.2</td>
</tr>
<tr>
<td>HAMSTER-weak-noeval</td>
<td>69.3</td>
</tr>
<tr>
<td>HAMSTER-original-test</td>
<td>67.1</td>
</tr>
<tr>
<td>HAMSTER-original-train</td>
<td>65.7</td>
</tr>
</tbody>
</table>

There are two baselines in Table 2: the first is the original performance of AMALGr as reported in Schneider, Danchik, et al. (2014) using CMWE (version 1.0 of this annotation), and the second is its performance using STREUSLE (version 3.0). Note that these involve exactly the same texts: the difference between these two numbers reflects other fixes to this dataset that have happened in the

\textsuperscript{3}The key difference between the CMWE and STREUSLE is the inclusion of supersense tags. Though we hope to eventually include supersense information in the output of HAMSTER, supersenses are beyond the scope of the present work.
years since its initial release. The difference between the two is quite substantial, at roughly 5% F-score.

The rest of the table makes use of HAMSTERized versions of STREUSLE, which we refer to as simply HAMSTER. The options here mostly refer to our treatment of the hard cases, which must be removed to make use of AMALGr. -original indicates that we apply all fixes which result in the creation or removal of a standard STREUSLE label (i.e., weak and strongish), but leave hard annotations as they were in the original corpus. -notMWE and -weak create versions of the corpus where all hard labels have been mapped to either nothing (no MWE) or weak MWEs, respectively. Another option we consider is -noeval, which involved tweaking the AMALGr evaluation script to exclude particular annotations (in this case hard) from evaluation altogether; that is, it does not matter what the model predicted for those words which are considered hard. Finally, -test and -train refer to the situation where we apply our fixes to texts only in the test or training sets, respectively; this gives us a sense of whether the improved performance of the model over the HAMSTER datasets is primarily due to the removal of errors from the test set, or whether improving the consistency of the training set is playing a major role as well.

Our fixes result in roughly another 5% increase to F-score relative to STREUSLE 3.0, for a total of about 10% F-score difference relative to results using the original CMWE annotation of this corpus. With respect to options for phrases labeled as hard, treating them as nonMWEs seems to be a worse option than simply leaving them alone; the best explanation for this is probably that these hard cases are generally more similar to labelled MWEs. Treating them as weak appears to a better strategy. Even better, though, might be to leave hard inconsistencies in the training set but exclude them from consideration during testing. The results using mixed training/test datasets indicate that the fixes to the test data are clearly more important, but the consistency across the two sets also accounts for a major part of the performance increase seen here.

Our second round of experiments looks at exact match recall with respect to various subsets of the MWEs in the test set. Here we consider only the original STREUSLE and HAMSTERized version with hard MWEs unchanged. N is the number of MWEs labeled as that type in that version of the dataset. Our goal here is to get a sense of how our changes have affected the identification of specific kinds of MWE. Weak versus strongish is an obvious distinction (mixed MWE were considered strongish), but even more relevant to what we have done here is whether or not the MWE appears in both the training and test sets. We are also interested in the status of multiword named entities (identified fairly reliably
using proper noun tags in the gold-standard POS tags), which occur numerously
in a corpus of reviews, but often as singletons, i.e., with a frequency of one. We
would expect MWEs which neither appear in our corpus nor are named entities
(NEs) to be relatively unaffected by our fixes, and among the most challenging
MWEs to identify in general.

Table 3: AMALGr exact recall for different MWE subsets in original
and HAMSTERized STREUSLE.

<table>
<thead>
<tr>
<th>MWE types</th>
<th>STREUSLE</th>
<th>HAMSTER</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>𝑁</td>
<td>Recall (%)</td>
</tr>
<tr>
<td>All</td>
<td>423</td>
<td>59.7</td>
</tr>
<tr>
<td>strongish</td>
<td>352</td>
<td>63.2</td>
</tr>
<tr>
<td>weak</td>
<td>71</td>
<td>24.0</td>
</tr>
<tr>
<td>In training</td>
<td>178</td>
<td>77.7</td>
</tr>
<tr>
<td>Not in training</td>
<td>247</td>
<td>47.4</td>
</tr>
<tr>
<td>Named entity (NE)</td>
<td>52</td>
<td>73.5</td>
</tr>
<tr>
<td>Not NE, not in training</td>
<td>195</td>
<td>40.3</td>
</tr>
</tbody>
</table>

In Table 3, AMALGr does better with the HAMSTER dataset for most of the
MWE subtypes considered here. The most striking difference occurs for the weak
tag, reflecting a disproportionate amount of inconsistency, enough that the model
built on the earlier version was apparently hesitant to apply the tag at all. Not
only are MWEs with training instances tagged better after our fixes, but the set
of such MWE tokens has noticeably increased. There is a corresponding drop in
those test instances without training data, which are clearly the most difficult
to identify, particularly when named entities are excluded. The recall of named
entities has actually dropped slightly, though since there are only 52 of these
in the test set, this corresponds to a single missed example and is probably not
meaningful. Though the rationale in terms of higher-level semantics is clear, we
wonder whether including NER as part of MWE identification may result in a
distorted view of the importance of MWE lexicons in token-level MWE iden-
tification. Here, we can see that among test-set-only MWEs, they stand out as
being significantly easier than the rest, probably because in English they can be
identified fairly reliably using only capitalization.
6 Discussion

Our three heuristics are useful because they identify potential errors with a high degree of precision. For the MWE type consistency analysis, 77% of candidate types were problematic, and for parse type consistency, the number was 63%. For the arc count heuristic, 54% of candidate types were ultimately changed: as mentioned earlier, many of the breaches involved systematic issues with annotation schema that we felt uncomfortable changing in isolation. By bringing these candidate instances to our attention, we were able to better focus our manual analysis effort, including in some cases looking across multiple related types, or even searching for specialist knowledge which could resolve ambiguities: for instance, in the example shown in Figure 1, though a layperson without reference material may be unsure whether it is tissue or massage which is considered to be deep, a quick online search indicates that the original EWT syntax is in error (deep modifies tissue).

However, it would be an overstatement to claim to have fixed all (or even almost all) the errors in the corpus. For instance, our type consistency heuristics only work when there are multiple instances of the same type, yet it is worth noting that 82% of the MWE types in the corpus are represented by a singleton instance. Our arc count heuristic can identify issues with singletons, but its scope is fairly limited. We cannot possibly identify missing annotations for types that were not annotated at least once. We might also miss certain kinds of systematic annotation errors, for instance those mentioned in De Smedt et al. (2015), though that work focused on the use of MWE dependency labels which are barely used in the EWT, one of the reasons a resource like STREUSLE is so useful.

Our experiments with the AMALGr tool show that our fixes result in a major improvement in MWE identification. One particularly striking result is the fact that the errors identified in the annotation since its original release account for about a quarter of all error (as measured by F-score) in the original model trained on it. This error may affect relative comparisons between systems, and we should be skeptical of results previously drawn based on relatively small differences in MWE identification in earlier versions of the corpus (e.g., Qu et al. 2015). This amount of error is also unacceptable simply in terms of the obfuscation relative to the degree of absolute progress on the task. Beyond this specific effort, we believe, for annotation efforts in general and for MWEs in particular, we should move beyond a singular focus on achieving sufficient annotator agreement in the initial annotation – the agreement in the original CWME was impressively high — and instead develop protocols for semi-automated, type-level inconsistency detection as a default step before any annotation is released.
7 Conclusion

We have proposed a methodology for merging MWE and dependency parse annotations, to generate HAMSTER: a gold-standard MWE-annotated dependency treebank with high consistency. The heuristics used to enforce consistency operate at the type- and cross-annotation level, and affected well over 10% of the MWEs in the new resource, resulting in a downstream change in MWE identification of roughly 5% F-score. More generally, we have provided here a case study in how bringing together multiple kinds of annotation done over the same corpus can facilitate rigorous error correction as part of the harmonization process.

Abbreviations

AMALGT A Machine Analyzer of Lexical Groupings
CMWE The Comprehensive Multiword Expression Corpus
EWT The English Web Treebank
HAMSTER The Harmonized Multiword and Syntactic Tree Resource
MWE multiword expression
NE named entity
NER named entity recognition
NLP Natural Language Processing
STREUSLE Supersense-Tagged Repository of English with a Unified Semantics for Lexical Expressions

References

Julian Brooke, King Chan & Timothy Baldwin


Semi-automated resolution of annotation inconsistency


Simi, Maria, Simonetta Montemagni & Cristina Bosco. 2015. Harmonizing and merging Italian treebanks: Towards a merged Italian dependency treebank and beyond. In Roberto Basili, Cristina Bosco, Rodolfo Delmonte, Alessandro Moschitti & Maria Simi (eds.), Harmonization and development of resources and tools for Italian Natural Language Processing within the PARLI project, 3–23. Heidelberg, Germany: Springer. DOI:10.1007/978-3-319-14206-7_1
