Phrase-Level Segmentation and Labelling of Machine Translation Errors

Frédéric Blain, Varvara Logacheva, Lucia Specia
Department of Computer Science
University of Sheffield
Sheffield, UK
{f.blain,v.logacheva,l.specia}@sheffield.ac.uk

Abstract

This paper presents our work towards a novel approach for Quality Estimation (QE) of machine translation based on sequences of adjacent words, the so-called phrases. This new level of QE aims to provide a natural balance between QE at word and sentence-level, which are either too fine grained or too coarse levels for some applications. However, phrase-level QE implies an intrinsic challenge: how to segment a machine translation into sequence of words (contiguous or not) that represent an error. We discuss three possible segmentation strategies to automatically extract erroneous phrases. We evaluate these strategies against annotations at phrase-level produced by humans, using a new dataset collected for this purpose.

Keywords: Machine Translation, Post-Editing, Quality Estimation

1. Introduction

We recently started to investigate Quality Estimation (QE) for Machine Translation (MT) at phrase-level (Logacheva and Specia, 2015) as a way to balance between word and sentence-level prediction, two well studied levels. Sentence-level QE generally aims to predict if a translation is either good enough or needs to be edited (and sometimes how much editing it needs). This is too coarse for certain tasks, for example, highlighting errors that need to be fixed. Word-level QE can help post-editors by highlighting words with errors, however, it is often hard to predict if an individual word is erroneous. Errors are generally interconnected within a segment, and it would be more beneficial for a post-editor if words belonging to the same instance of error could be grouped together, particularly for discontinuous errors, such as words in incorrect positions. However, contrary to the word-level QE, for which the segmentation boundaries are self-defined and clear, QE at phrase-level implies that one needs to delimit sub-segments within the segment. This is not a trivial task as several alternatives can be used to define a phrase, but in our case the segmentation needs to be connected to the errors in the translation.

QE at the phrase-level can reduce human post-editing effort by pinpointing the erroneous sequences that need to be fixed by the post-editor. It can also support automatic post-editing systems (McKeown et al., 2012; Chatterjee et al., 2015), by limiting the post-editing to sequences predicted as incorrect, and thus preventing risky edits that can make the translation even worse. The interest for automatic phrase-level segmentation (and labelling) is however not limited to QE. Given a human post-edition and its original machine translation, the combination of a monolingual alignment technique with an appropriate phrase segmentation would allow a more detailed analysis of the translation errors.

In Section 2. we discuss three possible ways of automatically segmenting a translation into phrases and labelling them with binary labels: "OK" or "BAD", with the latter indicating an error. In order to evaluate these strategies, we propose a new gold-standard resource built based on human annotations. The details of both the data collection experiment and the resulting dataset are given in Section 3. The results of our segmentation and labelling strategies against the gold-standard annotations are given in Section 4.

2. Segmentation & Labelling Strategies

2.1. Sentence Segmentation strategies

The definition of phrase differs depending on the task: in Linguistics, phrase is a unit where words are connected by dependency relationships. In Statistical MT (SMT), phrases are simply chains of words that frequently co-occur and are aligned with the same source word sequences. Therefore, we experimented with three segmentation strategies:

S1: Phrases from edit distance metric

Our first insight in terms of segmentation was to mimic as much as possible annotators' behaviour by producing a monolingual alignment between the raw machine translation and its post-edited version. We thus extract the phrases based on the edit path between these two sentences. Concretely, we first label as "BAD" every word in the MT output which has been marked as edited (inserted, substituted or moved) by the TERCOM tool (Snover et al., 2006). All remaining words are labelled as "OK". This is the standard procedure used currently to automatically generate labelled data from MT output and its post-edited version for word-level QE (Bojar et al., 2015). We add to this process then defining the final "OK" and "BAD" phrases as sequences of adjacent "OK" and "BAD" labels.

Different from the two strategies described next, this strategy is guided by the word-level labels, rather than the types of errors, and as a side effect it produces particularly long phrases, especially for the "OK" sequences. Thus, even though a phrase may be correctly tagged as "BAD", we lose the information on actual error boundaries in cases of multiple translation errors which happen to be consecutive, but are independent from each other.

S2: Linguistically motivated phrases
A key component to make a translation correct in the target language is the use of words that, when put together, make a coherent sub-sequences of words. This is particularly true for morphologically rich languages. On this basis, our motivation is to make use of linguistic information to determine the sentence segmentation. Therefore, our phrases are extracted from the shallow syntactic structure of the sentence (Constant et al., 2011), the so-called chunks, based on TreeTagger (Schmid, 1994). In future work, we could use dependency structures to assess whether the labelling of a phrase is dependent on or influences other phrases to re-define error boundaries. These error dependencies are a well-known phenomenon in MT, as it has been identified in (Blain et al., 2011).

S3: Decoder phrases
This approach, described in the research on phrase-level QE of (Logacheva and Specia, 2015), considers phrases in the SMT sense: sequences of words which often occur together and can be translated as one instance. The idea is to reuse the phrase segmentation produced by the decoder, with two hypotheses: (i) MT errors are usually context-dependent, so by dealing with the whole phrase we provide the local context related to the choice of a given word in phrase-based SMT and can more easily detect a single error which spans over two or more words, (ii) detecting errors at this level could be directly useful for using phrase-level quality predictions as additional features in an SMT decoder.
Sentences could thus be simply segmented into phrases based on the phrases actually used by SMT system decoder. However, since in our case we take an existing corpus, we need to re-translate the sentences in this corpus to obtain the phrase segmentation. We suggest two strategies:

- The source sentence is decoded by a source-target SMT system in a way that the output should be identical to the automatic translation in the corpus (i.e., "forced decoding"). This yields the segmentation of both source and target sentences with a one-to-one correspondence of segments.

- The target sentence is decoded by a target-source SMT system with no constraints. This decoding generates only the target part of the segmentation, the source phrases are generated from all source words aligned to words of a given target phrase.

The first scenario has the following drawback: when we perform forced decoding using an phrase table that is not exactly the same as that of the original system, the given reference translation is likely to be unreachable. In other words, the system can lack phrase pairs that translate source phrases to the given reference phrases. Therefore, in order to deliver the phrase segmentations for the given data we use a phrase table trained on the sentences we are decoding. This approach yields translations for the majority of sentences. However, for some of them (around 20% sentences for the considered dataset), the references still cannot be reached. In these cases we consider every word as a separate phrase.

The second scenario is more flexible: it is able to generate a segmentation for all sentences. However, similarly to the source-target approach, it depends on the data, in particular, on the training data of the SMT system used for decoding. If the data used for the MT system training and the sentences we are going to decode belong to different domains, there will be little intersection between the MT system's phrase table and the decoded sentences. As a result, the vast majority of identified phrases will be one-word, which will reduce the phrase-level QE task to the word-level QE.
For the target-source decoding strategy we used an SMT system trained on the English-French part of Europarl corpus (Koehn, 2005), built based on the Moses toolkit with standard settings (Koehn et al., 2007). Since our gold-standard sentences come from the LIG corpus, which was drawn from WMT test sets of different years (news domain), the system we used for decoding can be considered in-domain.

2.2. Phrase Labelling

Our labelling strategy is based on comparing the MT sentences and their version post-edited by a human, as it is done for labelling of word-level QE training data. This is only possible for labelling datasets at "training time" or for evaluation / translation quality analysis. Another option would be to rely on humans to tag each phrase as "OK" or "BAD", but this is costly and time consuming for the scale of datasets necessary for QE (thousands of sentences).
As one would expect, except for the phrases based on edit distance between the MT output and its post-edited version, the phrases generated automatically do not often match exactly the sequences labelled by the post-editor (i.e. spans of words labelled as "BAD" by the post-editor). So a phrase can contain words with both "BAD" and "OK" labels, whereas we need a single label for the entire phrase. Therefore, in cases of ambiguous labelling we use one of three heuristics to define a phrase-level label:

- optimistic – if half or more of words have a label "OK", the phrase has the label "OK" (majority labelling). That labelling was intended to keep the original balance of "OK" and "BAD" tags.

- pessimistic – if 30% words or more have a label "BAD", the phrase has the label "BAD". This strategy can be used in cases when the number of "BAD" words is not large and/or when the 'optimistic' labelling eliminates too many of them. The percentage of errors was chosen in order to convert three-word phrases with one "BAD" word into "BAD" phrases.

- super-pessimistic – if any word in the phrase has a label "BAD", the whole phrase has the label "BAD". This strategy is motivated by the possibility of using phrase-level QE to support phrase-based MT decoding. At each step of the search process the decoder chooses a new phrase, and ideally the best candidate phrase should contain only correct words. If one of the words does not fit the context, the entire phrase should be considered unsuitable.

3. Data Collection

As mentioned above, we faced the lack of reference annotation to evaluate our segmentation strategies against. We thus designed an annotation experiment to collect manually labelled phrase-level annotations of translation errors. For that, we made use of the "LIG corpus", a post-editing corpus described in (Potet et al., 2012). It contains 10.8k French-English translations, their post-edited versions, and reference translations, i.e. tuples of the type:

<source sentence, raw translation, post-edited translation, reference translation>

We asked human annotators, all fluent English speakers, to annotate a set of 10-50-word sentences extracted from the LIG corpus. One translation at a time, they were asked to annotate "BAD" phrases following a set of annotation guidelines. We decided to focus on annotating "BAD" phrases only because it is much harder to define guidelines for the segmentation of correct translations into phrases. As a consequence, we would have made the task very hard for humans and very prone to disagreements on segmentations of both "OK" and "BAD" phrases. In addition, we are interested in detecting and analysing errors, and the segmentation is only a means to get to those errors.

At the end of our experiment, it is about 1k annotations of manually-labelled "BAD" phrases which have been collected over 400 raw machine translations (about 10k words). These annotations are available under a Creative Commons Attribution-ShareAlike (CC-BY-SA) license to support further work on this topic. We also provide part of our scripts to facilitate reuse of our stand-off annotations with the original content of the LIG corpus (which has to be downloaded separately). These resources can be downloaded at: www.dcs.shef.ac.uk/~lucia/resources.html

3.1. Annotation Guidelines

The annotators were asked to identify any ungrammaticalities or variations of meaning that led to incorrect translations. To do so, they compared raw machine translations against their post-edited version, reference and source sentences. The reference and source sentences were given to help annotators identify variations of meaning that should be considered acceptable, since most annotators also spoke the source language, French. More specifically, we asked them to annotate cases that are not:

- Accurate, i.e. the target sentence does not accurately reflect the source sentence because of addition or omission of words, words that are translated with incorrect meaning.
- Fluent, with issues related to the text form, i.e. spelling, or grammar issues including word form or word order.

In order to make annotations as consistent as possible, we provided the annotators a set of guidelines, which we summarise here:

i) annotate as a single "BAD" phrase any single word or sequence of adjacent words belonging to the same error type. Conversely, annotate as different "BAD" phrases any sequences of adjacent words which seem to result from different types of translation errors;

ii) annotate as a single "BAD" phrase any sequence of adjacent words which may result from different types of errors, but where distinguishing and annotating these errors independently is too complex or may result in overlapping annotations;

iii) annotate an order error (a.k.a. shift) between two phrases by selecting the smallest phrase and indicating where it should be (by adding the that position as a fragment). In case of an order error between two phrases with the same length, annotate the first phrase and the place where it should be;

iv) two annotations should never overlap each other. If two annotations partially overlap, split them out into two distinct annotations. If an annotation is completely enclosing another annotation, keep only the annotation corresponding to the largest phrase;

v) annotate a missing phrase by selecting the last and first characters of the left and right words surrounding the position where it should be, and by providing the missing phrase. In this case, the phrase should be labelled as "BAD_DEL".

Guideline have been refined after a test session and examples were provided with each rule.

3.2. Annotation Environment

The annotations have been done and collected using the BRAT RAPID ANNOTATION TOOL[1] (Stenetorp et al., 2012), which provides an on-line environment for collaborative text annotation. Each annotator was provided with a full pre-configured version of the tool, as well as access to the guidelines. Figures 1 and 2 give an overview of the BRAT user interface with an annotation example. Figure 1 shows the visualisation interface where the annotator identifies phrases corresponding to MT errors. Figure 2 shows the labelling of a selected phrase according to the guidelines: "BAD_DEL".

Stand-off Format

Annotations created with BRAT are stored in a stand-off [2] format. In other words, annotations are stored into separate text file, with the original data remaining unchanged. In our experiment, BRAT's stand-off output was configured as follows: each line contains one annotation, and each annotation is given an ID that appears first on the line, separated from the rest of the annotation by a single TAB character. For example, this is the stand-off output for the annotation example shown in Figures 1 and 2:

```
T1 BAD_Del 59 62 t a
#1 AnnotatorNotes T1 that
T2 BAD 140 146;156 159 values y ,
T3 BAD 228 231 the
T4 BAD 75 79 will
T5 BAD 84 86 be
T6 BAD_Del 108 111 , w
#3 AnnotatorNotes T6 together
```

[1] http://brat.nlplab.org/

[2] http://brat.nlplab.org/standoff.html

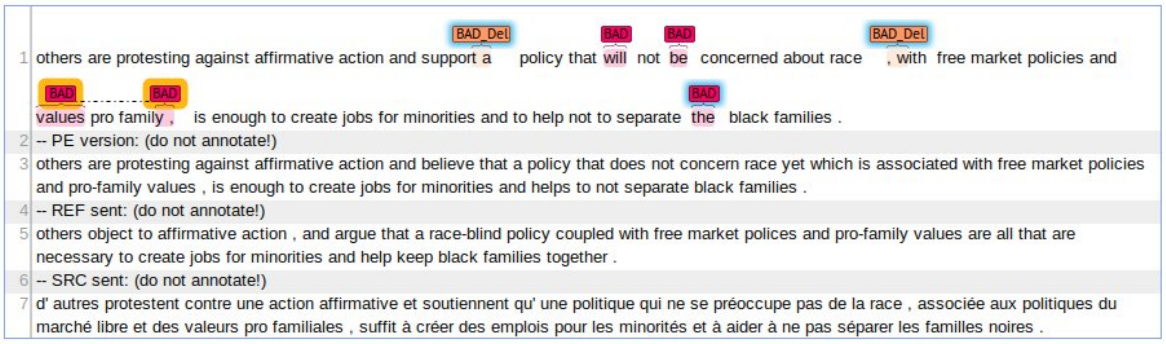

Figure 1: BRAT user interface for task visualisation. The sentence to annotate is displayed in the first line, the official post-edited and reference translations, as well as the source sentence, are given in the 3rd, 5th and 7th lines, respectively.

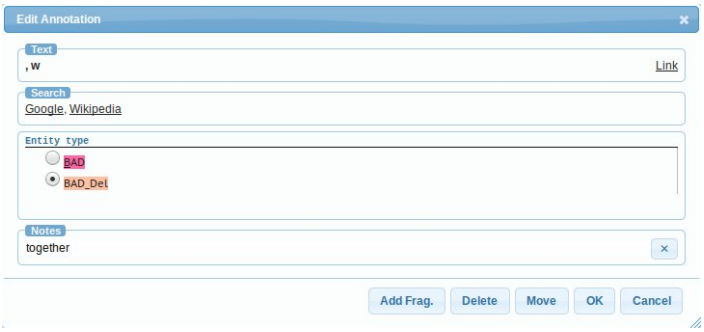

Figure 2: BRAT user interface for phrase annotation. Once the annotator has selected a particular phrase, they are asked to label it as either "BAD" or "BAD_DEL". For the latter case, the annotator also has to give the missing segment. Here the 1-word phrase "*together*" is annotated as missing at the position between the two tokens "," and "*with*".

In the next section, we make use of the new gold-standard dataset collected as described here to assess our three segmentation and labelling strategies.

4. Automatic vs. Gold-Standard Annotation

The *labelling* was evaluated in terms of F_1-score for the "BAD" class for phrases. This score is similar to one used for the evaluation of Named-Entity Recognition (NER) systems (Tjong Kim Sang and De Meulder, 2003). There, Precision is the percentage of named entities found by a system that are correct, Recall is the percentage of named entities present in the corpus that are found by a system, and F_1-score is the harmonic mean between these two metrics.

We could not evaluate the *segmentation* with F_1-score, because the Precision of the segments is meaningless in this case: since the annotators labelled only "BAD" phrases, most of the sentences contained a small number of phrases. In this scenario Precision will be inevitably low. Therefore, we evaluate segmentation in terms of Recall for the "BAD" phrases.

For both metrics we compute a strict and a relaxed version. The strict version counts only the exact matches between "BAD" phrases in the reference and the hypothesis, whereas the relaxed version takes into account the partial matches: if two "BAD" phrases overlap, their contribution to the overall score will be the ratio between the number of matching words and the length of this phrase in the reference sequence. Let us consider the following example:

```
Reference:  OK OK BAD OK BAD BAD OK
Hypothesis: OK OK BAD OK  OK  BAD OK
```

Here the gold standard has two "BAD" phrases. The hypothesis matches one of them exactly (3rd tag) and another one partially — while in the reference the 5th and 6th words are both "BAD", the hypothesis has only the 6th word marked with the "BAD" label. For the strict version of our metrics, we will take into account only the full match, and for the relaxed we will use both matches (in this case the total match count will be 1.5).

4.1. Phrase-Level Segmentation

Table 1 presents the results in terms of Recall of our segmentation strategies against our gold-standard data. We can observe that the edit-distance based approach performs better than the two others. This is not surprising since this strategy is based on the original post-editions. Therefore, this segmentation is close to the one produced by our annotators. Here, the difference between the actual score and its upper bound (100%) just mostly reflect the difference between the original post-editions provided in the LIG corpus and the errors identified during our annotation experiment. The very low strict Recall for the source-target decoder-based approach could be explained by the fact that we enriched the phrase-table of the SMT system with an additional phrase table trained on our data in order to avoid the lack of suitable phrase pairs. As a side effect, this resulted in much longer phrases.

4.2. Segmentation Labelling

The results of labelling with different heuristics are given in Table 2. While the strategy based on edit-distance got the best result in terms of strict F_1-score, the source-target decoder-based segmentation got the best result for the relaxed version. We notice the following regularity for all the segmentation strategies: as we go from "optimistic" labelling to "pessimistic" and then "super-pessimistic", our strict F_1-score decreases, while the relaxed score goes up.

Segmentation Strategy	Recall (%) Strict	Relaxed
Edit-Distance:		
phrase length up to 5	**42.49**	**87.84**
phrase length unlimited	42.02	84.15
Shallow Syntactic decomposition	33.17	82.79
Decoder-based:		
source-target SMT	27.12	81.69
target-source SMT	25.26	84.61

Table 1: Evaluation of our segmentation strategies in terms of Recall (strict and relaxed) against our gold-standard data.

Segmentation Strategy	F_1-score (%) Strict	Relaxed
Edit-Distance:		
phrase length up to 5	35.35	53.08
phrase length unlimited	**35.64**	**53.32**
– OPTIMISTIC LABELLING:		
Shallow Syntactic decomposition	**19.88**	33.98
Decoder-based:		
source-target SMT	17.66	32.86
target-source SMT	17.60	34.44
– PESSIMISTIC LABELLING:		
Shallow Syntactic decomposition	17.09	44.07
Decoder-based:		
source-target SMT	15.42	41.26
target-source SMT	14.56	46.86
– SUPER-PESSIMISTIC LABELLING:		
Shallow Syntactic decomposition	16.83	44.47
Decoder-based:		
source-target SMT	14.15	47.14
target-source SMT	14.26	**47.25**

Table 2: Evaluation of our labelling heuristics on our segmentations in terms of F_1-score for the "BAD" phrases against our gold-standard data.

The inflated relaxed score are explained by the fact that the "optimistic" labelling replaces many original "BAD" labels with "OK" labels. As we switch to "pessimistic" scheme, the number of "BAD" labels in the data increases which results in more partial matches. However, the strict score does not follow this pattern.

In order to understand the reason for the difference in scores behaviour, we explore the components which form the overall F_1-score: Precision and Recall. In our case Precision is the ratio between the number of "BAD" phrases that match exactly in the reference and hypothesis (True Positives (TP)) and the overall number of "BAD" phrases in the hypothesis (True Positives + False Positives (TP+FP)). Table 3 shows how these figures change when we move

Labelling Strategy	TP (#)	TP+FP (#)	Prec. (%)	Rec. (%)	F_1-score (%)
optimistic	186	1012	18.37	21.65	19.88
pessimistic	196	1434	13.66	22.81	17.09
super-pessimistic	196	1470	13.33	22.81	16.83

Table 3: The variation in the number of "BAD" phrases for different labelling strategies (for shallow syntactic decomposition segmentation).

to more pessimistic labellings for the shallow syntactic decomposition segmentation strategy (but the same regularities hold for other segmentation strategies as well). As we decrease the threshold of "BAD" labels percentage (*i.e.* raise the number of "BAD" phrases in the data), the number of matching phrases goes up slightly, but the increase in the overall number of "BAD" phrases is much more remarkable. Since the number of phrases in the reference does not change, the Recall grows marginally, but the drop in Precision is larger, and the final F_1-score is dominated by it.

We can notice an overall low F_1-score which suggests a significant disagreement between our automatic segmentations and the human annotators. Part of this gap could be explained by the fact that post-editors who produced the initial post-editions in (Potet et al., 2012) had the access only to the source sentences and their automatic translations, whereas for our experiments we gave the annotators the access to all the available data: source sentences, automatic translations, post-editions and reference translations, so they could decide on the optimal labelling from a range of possibilities including existing corrections and their own knowledge. Thus, where our phrase labelling would consider as "BAD" a phrase which has been modified in the post-edited version of the translation, a human annotator might consider the meaning as unchanged and therefore would not label this modification as an MT error.

5. Conclusions

Our experience in Quality Estimation led us to look at a novel approach based on sequences of adjacent words, so-called phrase, as a natural balance between the too fine grained word- and too coarse sentence-levels. However an intrinsic challenge comes along with this new level: how to find phrases which correspond to actual machine translation errors. While boundaries for both word- and sentence-level are self-defined, this is an open question for the intermediate level.

In this paper we presented three possible segmentation approaches: based on edit-distance, shallow syntactic decomposition and decoder segmentation. We also presented three labelling strategies to automatically extract the erroneous phrases from a post-editing corpus.

Additionally, we introduced a new dataset that we created for assessing our automatic strategies against. This dataset is the result of an annotation experiment done with the help of English speakers. It provides gold-standard phrase-level

annotations of machine translations errors. For this first version, we collected a set of 1k annotations over 400 sentences. In order to support further work, we made it available, as mentioned in Section 3.

The results reported in this paper represent the first step of our work on segmentation and labelling for this new level for Quality Estimation. They are promising, even though they show that our segmentation and labelling strategies need to be refined in future work.

Acknowledgements

The authors would like to thanks all the annotators who helped to create the first version of gold-standard annotations at phrase-level. This work was supported by the QT21 (H2020 No. 645452, Lucia Specia, Frédéric Blain) and EXPERT (EU FP7 Marie Curie ITN No. 317471, Varvara Logacheva) projects.

References

Blain, F., Senellart, J., Schwenk, H., Plitt, M., Roturier, J., and Blain. (2011). Qualitative analysis of post-editing for high quality machine translation. In Asia-Pacific Association for Machine Translation (AAMT), editor, *Machine Translation Summit XIII*, Xiamen (China), 19-23 sept.

Bojar, O., Chatterjee, R., Federmann, C., Haddow, B., Huck, M., Hokamp, C., Koehn, P., Logacheva, V., Monz, C., Negri, M., Post, M., Scarton, C., Specia, L., and Turchi, M. (2015). Findings of the 2015 workshop on statistical machine translation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 1–46, Lisboa, Portugal.

Chatterjee, R., Weller, M., Negri, M., and Turchi, M. (2015). Exploring the planet of the apes: a comparative study of state-of-the-art methods for mt automatic post-editing. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics), Beijing, China*.

Constant, M., Tellier, I., Duchier, D., Dupont, Y., Sigogne, A., and Billot, S. (2011). Intégrer des connaissances linguistiques dans un crf: application à l'apprentissage d'un segmenteur-étiqueteur du français. In *TALN*, volume 1, page 321.

Koehn, P., Hoang, H., Birch, A., Callison-burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open Source Toolkit for Statistical Machine Translation. In *ACL-2007: 45th Annual Meeting of the Association for Computational Linguistics, Demo and Poster sessions*, pages 177–180, Prague, Czech Republic.

Koehn, P. (2005). Europarl: A Parallel Corpus for Statistical Machine Translation. In *MT Summit X: 10th Machine Translation Summit*, pages 79–86, Phuket, Thailand.

Logacheva, V. and Specia, L. (2015). Phrase-level quality estimation for machine translation. In *Proceedings of IWSLT-2015*.

McKeown, K., Parton, K., Habash, N. Y., Iglesias, G., and de Gispert, A. (2012). Can automatic post-editing make mt more meaningful?

Potet, M., Esperança-Rodier, E., Besacier, L., and Blanchon, H. (2012). Collection of a large database of french-english smt output corrections. In *LREC*, pages 4043–4048. Citeseer.

Schmid, H. (1994). Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the international conference on new methods in language processing*, volume 12, pages 44–49. Citeseer.

Snover, M., Dorr, B., Schwartz, R., Micciulla, L., and Makhoul, J. (2006). A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas (AMTA)*, pages 223–231.

Stenetorp, P., Pyysalo, S., Topić, G., Ohta, T., Ananiadou, S., and Tsujii, J. (2012). brat: a web-based tool for NLP-assisted text annotation. In *Proceedings of the Demonstrations Session at EACL 2012*, Avignon, France, April. Association for Computational Linguistics.

Tjong Kim Sang, E. F. and De Meulder, F. (2003). Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL-2003*, pages 142–147.

Language Resource References

Marion Potet and Emmanuelle Esperança-Rodier and Laurent Besacier and Hervé Blanchon. (2012). *Collection of a Large Database of French-English SMT Output Corrections.*