Mach Translat (2018) 32:325–351
https://doi.org/10.1007/s10590-018-9218-6

Automatic quality estimation for speech translation using joint ASR and MT features

Ngoc-Tien Le[1] · **Benjamin Lecouteux**[1] · **Laurent Besacier**[1]

Received: 29 July 2016 / Accepted: 22 March 2018 / Published online: 1 June 2018

Abstract This paper addresses the automatic quality estimation of spoken language translation (SLT). This relatively new task is defined and formalized as a sequence-labeling problem where each word in the SLT hypothesis is tagged as *good* or *bad* according to a large feature set. We propose several word confidence estimators (WCE) based on our automatic evaluation of transcription (ASR) quality, translation (MT) quality, or both (combined ASR + MT). This research work is possible because we built a specific corpus, which contains 6.7k utterances comprising the quintuplet: ASR output, verbatim transcript, text translation, speech translation, and post-edition of the translation. The conclusion of our multiple experiments using joint ASR and MT features for WCE is that MT features remain the most influential while ASR features can bring interesting complementary information. In addition, the last part of the paper proposes to disentangle ASR errors and MT errors where each word in the SLT hypothesis is tagged as *good*, *asr_error* or *mt_error*. Robust quality estimators for SLT can be used for re-scoring speech translation graphs or for providing feedback to the user in interactive speech translation or computer-assisted speech-to-text scenarios.

Keywords Quality estimation · Word confidence estimation (WCE) · Spoken language translation (SLT) · Joint features · Feature selection

1 Introduction

Automatic quality assessment of spoken language translation (SLT), also named confidence estimation (CE), is an important topic because it allows us to know whether a

✉ Laurent Besacier
laurent.besacier@imag.fr

[1] Laboratoire d'Informatique de Grenoble, University of Grenoble Alpes, Building IMAG, 700 Centrale, 38401 Saint Martin d'Hères, France

system produces user-acceptable outputs or not. In interactive speech-to-speech translation, CE helps to judge whether a translated term is uncertain (in which case we can ask the speaker to rephrase or repeat the term). For speech-to-text applications, CE may tell us whether output translations are worth correcting, or whether they require retranslation from scratch. Moreover, an accurate CE can also help to improve SLT itself through a second-pass N-best list re-ranking or search graph re-decoding, as has already been done for text translation in Bach et al. (2011) and Luong et al. (2014b), or for speech translation in Besacier et al. (2015). Consequently, building a method which is capable of pointing out the correct parts as well as detecting the errors in a speech-translated output is crucial to tackle the above issues.

Given a signal x_f in the source language, spoken language translation (SLT) consists of finding the most probable target language sequence $\hat{e} = (e_1, e_2, \ldots, e_N)$, as in (1):

$$\hat{e} = \arg\max_{e} \left\{ p(e|x_f, f) \right\} \tag{1}$$

where $f = (f_1, f_2, \ldots, f_M)$ is the transcription of x_f. Now, if we perform confidence estimation at the "word" level, the problem is called word-level confidence estimation (WCE) and we can represent this information as a sequence q (same length N of $\hat{e}$) where $q = (q_1, q_2, \ldots, q_N)$ and $q_i \in \{good, bad\}$.[1]

Then, integrating automatic quality assessment into our SLT process (q is defined above) can be done as in (2)–(4):

$$\hat{e} = \arg\max_{e} \sum_{q} \left\{ p(e, q|x_f, f) \right\} \tag{2}$$

$$\hat{e} = \arg\max_{e} \sum_{q} \left\{ p(q|x_f, f, e) * p(e|x_f, f) \right\} \tag{3}$$

$$\hat{e} \approx \arg\max_{e} \{ \max_{q} \{ p(q|x_f, f, e) * p(e|x_f, f) \} \} \tag{4}$$

In the product of (4), the SLT component $p(e|x_f, f)$ and the WCE component $p(q|x_f, f, e)$ contribute together to find the best translation output $\hat{e}$. In the past, WCE has been treated separately in ASR or MT contexts and we propose here a joint estimation of word confidence for an SLT task involving both ASR and MT.

This journal paper is an extended version of a paper published at ASRU 2015 (Besacier et al. 2015), but here we focus more on the WCE component and on the best approaches to accurately estimate $p(q|x_f, f, e)$.

Contributions A corpus (distributed to the research community)[2] dedicated to WCE for SLT was initially published in Besacier et al. (2014). In this paper, we present its extension from 2643 to 6693 speech utterances. In addition, while our previous work on quality assessment was based on two separate WCE classifiers (one for quality assessment in ASR and one in MT), we propose here a unique *joint* model based

[1] q_i could be also more than 2 labels, or even scores, but this paper mostly deals with error detection using a binary set of labels, with the exception of Sect. 7 where three labels are considered.

[2] https://github.com/besacier/WCE-SLT-LIG.

on different feature types (ASR and MT features). This *joint* model allows us to operate feature selection and analyze which features (from ASR or MT) are the most efficient for quality assessment in speech translation. We also experiment with two ASR systems that have different performance in order to analyze the behaviour of our SLT quality-assessment algorithms at different levels of word error rate (WER) (Levenshtein 1966). The last part of this paper proposes to disentangle ASR and MT errors in speech translation by automatically detecting the origin of SLT errors, either due to ASR or to MT.

Outline The outline of this paper is as follows: Sect. 2 reviews the state-of-the-art on confidence estimation for ASR and MT. Our word-confidence estimation (WCE) system using multiple features is then described in Sect. 3. The experimental setup (namely our specific WCE corpus) is presented in Sect. 4 while Sect. 5 evaluates our joint WCE system. Feature selection for quality assessment in speech translation is analyzed in Sect. 6. Section 7 proposes to disentangle ASR and MT errors in SLT output and finally, Sect. 8 concludes this work and gives some future perspectives.

2 Related work on confidence estimation for ASR and MT

Several previous works tried to propose effective confidence measures in order to detect errors on ASR outputs. Confidence measures are introduced for Out-Of-Vocabulary (OOV) detection by Asadi et al. (1990). Young (1994) extends this previous work and introduces the use of word posterior probability (WPP) as a confidence measure for speech recognition. The posterior probability of a word is most of the time computed using the hypothesis word graph (Kemp and Schaaf 1997). More recent approaches (Lecouteux et al. 2009) for confidence measure estimation use side-information extracted from the recognizer: normalized likelihoods (WPP), the number of competitors at the end of a word (hypothesis density), decoding process behaviour, linguistic features, acoustic features (acoustic stability, duration features), and semantic features. In addition, ASR quality estimation has been addressed in several recent studies (de Souza et al. 2015; Zamani et al. 2015; Jalalvand et al. 2016). Re-scoring the output of the ASR N-best list using ASR and MT features was also proposed by Ng et al. (2014, 2015, 2016).

In parallel, the Workshop on Machine Translation (WMT)[3] introduced in 2013 a WCE task for Machine Translation. Han et al. (2013) and Luong et al. (2013b) employed the Conditional Random Fields (CRF) (Lafferty et al. 2001) model as their machine-learning method to address the problem as a sequence-labelling task. Meanwhile, Biçici (2013) extended their initial proposition by dynamic training with adaptive weight updates in their neural network classifier. As far as prediction indicators are concerned, Biçici (2013) proposed seven word feature types and found among them the "common cover links" (the links that point from the leaf node containing this word to other leaf nodes in the same subtree of the syntactic tree) the most outstanding. Han et al. (2013) focused only on various N-gram combinations of target words. Inheriting most of the previously recognized features, Luong et al. (2013b)

[3] cf. http://www.statmt.org/wmt17/ for the most recent such instance.

integrated a number of new indicators relying on graph topology, pseudo reference, syntactic behaviour (constituent label, distance to the semantic tree root) and polysemy characteristic. The estimation of the confidence score mainly uses classifiers like CRF (Han et al. 2013; Luong et al. 2014a), Support Vector Machines (Langlois et al. 2012), or neural networks (Biçici 2013). Some investigations were also conducted to determine which features seem to be the most relevant. Langlois et al. (2012) proposed to filter features using a forward-backward algorithm to discard linearly correlated features. Using boosting as the learning algorithm, Luong et al. (2015) were able to take advantage of the most significant features.

Finally, several toolkits for WCE have recently been proposed: *TranscRater* for ASR (Jalalvand et al. 2016),[4] QuEst++ for MT (Specia et al. 2015),[5] MARMOT for MT (Logacheva et al. 2016),[6] as well as the WCE toolkit (Servan et al. 2015)[7] that is used to extract MT features in the experiments of this paper.

To the best of our knowledge, the first attempt to design WCE for speech translation, using both ASR and MT features in a single classifier, was our own work (Besacier et al. 2014, 2015) which is further extended in this paper.

3 Building an efficient quality assessment (WCE) system

The WCE component solves equation (5):

$$\hat{q} = \arg\max_{q} \{p_{SLT}(q|x_f, f, e)\} \tag{5}$$

where $q = (q_1, q_2, \ldots, q_N)$ is the sequence of quality labels on the target language. This is a sequence-labelling task that can be solved with several machine-learning techniques such as CRF. However, for that, we need a large amount of training data for which a quadruplet (x_f, f, e, q) is available. In this work, we will use a corpus extended from Besacier et al. (2014) which contains 6.7k utterances. We will investigate if this amount of data is enough to evaluate and test a joint model $p_{SLT}(q|x_f, f, e)$.

As it is much easier to obtain data containing either the triplet (x_f, f, q) (automatically transcribed speech with manual references and quality labels inferred from WER estimation), or the triplet (f, e, q) (automatically translated text with manual post-edits and quality labels inferred using tools such as TERp-A (Snover et al. 2009)), we can also recast the WCE problem as in (6):

$$\hat{q} = \arg\max_{q} \{p_{ASR}(q|x_f, f)^{\alpha} * p_{MT}(q|e, f)^{1-\alpha}\} \tag{6}$$

where α is a weight giving more or less importance to WCE_{ASR} (quality assessment on transcription) compared to WCE_{MT} (quality assessment on translation). It is important

[4] https://github.com/hlt-mt/TranscRater.

[5] http://www.quest.dcs.shef.ac.uk/.

[6] https://github.com/qe-team/marmot.

[7] https://github.com/besacier/WCE-LIG.

to note that $p_{ASR}(q|x_f, f)$ corresponds to the quality estimation of the words in the target language based on features calculated on the source language (ASR). For that, what we do is project source quality scores to the target using word alignment information between e and f sequences. This alternative approach (Eq. (6)) will be also evaluated in this work even if it corresponds to a different optimization problem than Eq. (5). In particular, the choice of α is only set *a priori* in our experiments (to 0.5) which is probably not the best option.

In both approaches—*joint* ($p_{SLT}(q|x_f, f, e)$) and *combined* ($p_{ASR}(q|x_f, f)$ + $p_{MT}(q|e, f)$)—some features need to be extracted from ASR and MT modules. They are more precisely detailed in the next subsections.

3.1 WCE features for speech transcription (ASR)

In this work, we extract several types of features, which come from the ASR graph, from language model scores and from a morphosyntactic analysis. These features are listed below [more details can be found in Besacier et al. (2014)]:

- *Acoustic features* word duration (**F-dur**).
- *Graph features (extracted from the ASR word confusion networks)* number of alternative (**F-alt**) paths between two nodes; word posterior probability (**F-post**).
- *Linguistic features (based on probabilities by the language model)* the word itself (**F-word**), 3-gram probability (**F-3g**), log probability (**F-log**), back-off level of the word (**F-back**), as proposed in Fayolle et al. (2010).
- *Lexical Features* Part-Of-Speech (POS) of the word (**F-POS**).
- *Context Features* Part-Of-Speech tags in the neighborhood of a given word (**F-context**).

For each word in the ASR hypothesis, we estimate these 9 features: F-Word; F-3g; F-back; F-log; F-alt; F-post; F-dur; F-POS; and F-context.

In a preliminary experiment, we evaluate these features for quality assessment in ASR only (WCE_{ASR} task). Two different classifiers will be used: a variant of boosting classification algorithm called *bonzaiboost* (Laurent et al. 2014), which implements the boosting algorithm *Adaboost.MH* over deeper trees, and CRF.

3.2 WCE features for machine translation (MT)

A number of knowledge sources are employed for extracting features, in a total of 24 major feature types, as depicted in Table 1.

It is important to note that we extract features regarding *tokens* in the translated hypothesis (MT or SLT). In other words, one feature is extracted for each token in the MT output. So in Table 1, *target* refers to the feature coming from the translated hypothesis and *source* refers to a feature extracted from the source word aligned to the considered target word. More details on some of these features are given in the next subsections.

Table 1 List of MT features extracted

1. Proper name	10. Stop word	19. WPP max
2. Unknown stem	11. Word context alignments	20. Nodes
3. Num. of word occ.	12. POS context alignments	21. Constituent label
4. Num. of stem occ.	13. Stem context alignments	22. Distance to root
5. Polysemy count—target	14. Longest target N-gram length	23. Numeric
6. Backoff behaviour—target	15. Longest source N-gram length	24. Punctuation
7. Alignment features	16. WPP exact	
8. Occur in google translate	17. WPP any	
9. Occur in Bing translate	18. WPP min	

3.2.1 Internal features

These features are given by the MT system, which outputs additional data like an N-best list.

Word Posterior Probability (WPP) and **Nodes** features are extracted from a confusion network, which comes from the output of the MT N-best list. **WPP Exact** is the WPP value for each word concerned at the exact same position in the graph. **WPP Any** extracts the same information at any position in the graph. **WPP Min** gives the smallest WPP value concerned by the transition and **WPP Max** its maximum.

3.2.2 External features

Below is the list of the external features used:

- *Proper name* indicates if a word is a proper name; the same binary features are extracted to know if a token is **Numerical**, **Punctuation**, or a **Stop Word**.
- (*Unknown stem* informs whether the stem of the considered word is known or not.
- *Number of word/stem occurrences* counts the occurrences of a word/stem in the sentence.
- *Alignment context features* these features (#11–13 in Table 1) are based on collocations and were proposed by Bach et al. (2011). Collocations could be an indicator for judging whether a target word is generated by a particular source word. We also apply the reverse, the collocations regarding the source side (#7 in Table 1, simply called **Alignment Features**):
  - ◇ *Source alignment context features* the combinations of the target word, the source word (with which it is aligned), and one source word before and one source word after (left and right contexts, respectively).
  - ◇ *Target alignment context features* the combinations of the source word, the target word (with which it is aligned), and one target word before and one target word after.
- *Longest target (or source) N-gram length* we seek to compute the length $(n + 1)$ of the longest left sequence (w_{i-n}) concerned by the current word (w_i) and known by the language model (LM) concerned (source and target sides). For example,

if the longest left sequence w_{i-2}, w_{i-1}, w_i appears in the target LM, the longest target N-gram value for w_i will be 3. This value ranges from 0 to the max order of the LM concerned. We also extract a redundant feature called **Backoff Behaviour Target**.

- The target word's constituent label (**Constituent Label**) and its depth in the constituent tree (**Distance to Root**) are extracted using a syntactic parser.
- *Target polysemy count* we extract the polysemy count, which is the number of meanings of a word in a given language.
- *Occurrences in google translate* and *Occurrences in bing translator* in the translation hypothesis, we (optionally) test the presence of the target word in on-line translations given respectively by *Google Translate* and *Bing Translator*.[8]

A very similar feature set was used for a simple WCE_{MT} task (English–Spanish MT, WMT 2013, 2014 quality-estimation shared task) and obtained very good performance (Luong et al. 2013a). This preliminary experience in participating to the WCE shared task in 2013 and 2014 led us to the following observation: while feature processing is very important to achieve good performance, it requires a set of heterogeneous NLP tools (for lexical, syntactic, and semantic analysis). Thus, we recently proposed to unify the feature processing, together with the call of machine-learning algorithms, in order to facilitate the design of confidence-estimation systems. The open-source toolkit proposed (written in *Python* and made available on *github*)[9] integrates some standard as well as in-house features that have proven useful for WCE (based on our experience in WMT 2013 and 2014).

In this paper, we use only CRF as our machine-learning method, with the WAPITI toolkit (Lavergne et al. 2010), to train our WCE estimator based on both MT and ASR features.

4 Experimental setup

4.1 Dataset

4.1.1 Starting point: an existing MT post-edition corpus

For a French–English translation task, we used our SMT system to obtain the translation hypotheses for 10,881 source sentences taken from news corpora from WMT evaluation campaigns from 2006 to 2010. Post-editions were obtained from non professional translators using a crowdsourcing platform. More details on the baseline SMT system used can be found in Potet et al. (2010), and more details on the post-edited corpus can be found in Potet et al. (2012). It is worth mentioning, however, that a subset (311 sentences) of these collected post-editions was assessed by a professional translator and 87.1% of the post-edits were judged to improve the hypothesis.

[8] Using this kind of feature is controversial, but we observed that such features are available in general scenarios, so we decided to include them in our experiments. Contrastive results without these two features will be also given later on.

[9] http://github.com/besacier/WCE-LIG.

Table 2 Example of training label obtained using TERp-A

Reference	The	consequence	of	the	fundamentalist	
	E	S	E	E	S	
Hyp after shift	The	result	of	the	hard-line	
Reference	movement		also	has	its importance	
	Y	I	E	D	P	E
Hyp after shift	trend	is	also		important	

Table 3 Details on our *dev* and *tst* corpora for SLT

Corpus	#Sentences	#Speech recordings	#Speakers	Duration
dev	881	2643	15 (9 women + 6 men)	5h51
tst	1350	4050	27 (11 women + 16 men)	11h01

Then, the word-label setting for WCE was done using the TERp-A toolkit (Snover et al. 2009). Table 2 illustrates the labels generated by TERp-A for one hypothesis and post-edition pair. Each word or phrase in the hypothesis is aligned to a word or phrase in the post-edition with different types of edit operations: "I" (insertions), "S" (substitutions), "T" (stem matches), "Y" (synonym matches), and "P" (phrasal substitutions). The lack of a symbol indicates an exact match and will be replaced by "E" thereafter. We do not consider words marked with "D" (deletions) since they appear only in the reference. However, later on, we will have to train binary classifiers ($good/bad$) so we re-categorize the obtained 6-label set into a binary set: E, T and Y belong to the *good* (G) class, whereas S, P and I belong to the *bad* (B) category.

4.1.2 Extending the corpus with speech recordings and transcripts

The *dev* set and *tst* set of this corpus were recorded by French native speakers. Each sentence was uttered by 3 speakers, leading to 2643 and 4050 speech recordings for the *dev* set and *tst* set, respectively. For each speech utterance, a quintuplet containing ASR output (f_{hyp}), verbatim transcript (f_{ref}), English text-translation output ($e_{hyp_{mt}}$), speech-translation output ($e_{hyp_{slt}}$) and post-edition of translation (e_{ref}) was made available. This corpus is available on a *github* repository.[10] More details are given in Table 3. The total length of the *dev* and *tst* speech corpora obtained are 16h52, since some utterances were pretty long.

4.2 ASR systems

To obtain the speech transcripts (f_{hyp}), we built a French ASR system based on KALDI toolkit (Povey et al. 2011). Acoustic models are trained using several corpora (ESTER,

[10] https://github.com/besacier/WCE-SLT-LIG/.

Table 4 Details on language models (LMs) used in our two ASR systems

LM	1-g	2-g	3-g
Small ($ASR1$)	62K	1M	59M
Big ($ASR2$)	95K	49M	301M

Table 5 ASR performance (WER) on our *dev* and *tst* set for the two different ASR systems

Task	*dev* set (%)	*tst* set (%)
$ASR1$	21.86	17.37
$ASR2$	16.90	12.50

REPERE, ETAPE and BREF120) representing more than 600 h of French transcribed speech.

The baseline GMM system is based on mel-frequency cepstral coefficient (MFCC) acoustic features (13 coefficients expanded with delta and double delta features and energy: 40 features) with various feature transformations including linear discriminant analysis (LDA), maximum likelihood linear transformation (MLLT), and feature space maximum likelihood linear regression (fMLLR) with speaker adaptive training (SAT). The GMM acoustic model makes initial phoneme alignments of the training data set for the following DNN acoustic model training.

The speech transcription process is carried out in two passes: an automatic transcript is generated with a GMM-HMM model of 43,182 states and 250,000 Gaussians. Then word-graph outputs obtained during the first pass are used to compute a fMLLR-SAT transform on each speaker. The second pass is performed using DNN acoustic model trained on acoustic features normalized with the fMLLR matrix.

CD-DNN-HMM acoustic models are trained (43,182 context-dependent states) using a GMM-HMM topology.

We propose to use two 3-g language models trained on the French ESTER corpus (Galliano et al. 2006) as well as on French Gigaword (vocabulary sizes are 62 and 95k, respectively). The LM weight parameters of the ASR systems are tuned via WER on the *dev* corpus. Details on these two language models can be found in Table 4.

In our experiments, we propose two ASR systems based on the previously described language models. The first system ($ASR1$) uses the small language model allowing a fast ASR system (about 2× Real Time), while in the second system lattices are rescored with a big language model (about 10× Real Time) during a third pass.

Table 5 presents the performances obtained by two above ASR systems.

These WER scores may appear rather high for the task of transcribing read news. A deeper analysis shows that these news items contain a lot of foreign named entities, especially in our *dev* set. This part of the data is extracted from French media dealing with the European economy. This could also explain why the scores are significantly different between the *dev* and *tst* sets. In addition, automatic post-processing is applied to the ASR output in order to match the requirements of standard input for MT.

4.3 SMT system

We used the Moses phrase-based translation toolkit (Koehn et al. 2007) to translate French ASR into English (e_{hyp}). This medium-sized system was trained using a subset of data provided for IWSLT 2012 evaluation (Federico et al. 2012): Europarl, Ted and News-Commentary corpora, with a total amount of about 60M words. We used an adapted target language model trained on specific data (News Crawled corpora) similar to our evaluation corpus (see Potet et al. (2010)). This standard SMT system is used in all experiments reported in this paper (Tables 6, 7).

4.4 Obtaining quality assessment labels for SLT

After building an ASR system, we have a new element of our desired quintuplet: the ASR output f_{hyp}, which is the noisy version of our already available verbatim transcripts called f_{ref}. This ASR output (f_{hyp}) is then translated by the exact same SMT system (Potet et al. 2010) mentioned in Subsect. 4.3. This new output translation is called $e_{hyp_{slt}}$ and is a degraded version of $e_{hyp_{mt}}$ (translation of f_{ref}).

At this point, a strong assumption we made has to be revealed: we re-used the post-editions obtained from the text-translation task (called e_{ref}), to infer the quality (G, B) labels of our speech translation output $e_{hyp_{slt}}$. The word-label setting for WCE is done using TERp-A toolkit (Snover et al. 2009) between $e_{hyp_{slt}}$ and e_{ref}. This assumption, and the fact that initial MT post-edition can also be used to infer labels of a SLT task, is reasonable regarding results (presented later in Tables 8, 9) where it is shown that there is not a huge difference between the MT and SLT performance (evaluated with BLEU) (Table 10).

The above remark is important and this is what makes this corpus valuable. For instance, other corpora such as the TED corpus could be used to obtain a quintuplet with ASR output, verbatim transcript, MT output, SLT output and target translation, but there are two main differences: first in TED, the target translation is a manual translation of the prior subtitles so this is not a post-edition of an automatic translation (and we have no guarantee that the *good*/*bad* labels extracted from this would be reliable for WCE training and testing); secondly, in our corpus, each sentence is uttered by 3 different speakers which introduces speaker variability in the database and allows us to deal with different ASR outputs for a single source sentence.[11]

4.5 Final corpus statistics

The final corpus obtained is summarized in Table 6, where we also clarify how the WCE labels were obtained. For the test set, we now have all the data needed to evaluate WCE for 3 tasks:

- *ASR* extract *good*/*bad* labels by calculating WER between f_{hyp} and f_{ref},

[11] These 3 alternative utterances are simply added to the corpus as 3 examples and are used independently of each other.

Table 6 Overview of our post-edition corpus for SLT

Data	#*dev* utt	#*tst* utt	#*dev* words	#*tst* words	Method to obtain WCE labels
f_{ref}	881	1350	21,988	36,404	
f_{hyp1}	881*3	1350*3	66,435	108,332	wer(f_{hyp1}, f_{ref})
f_{hyp2}	881*3	1350*3	66,834	108,598	wer(f_{hyp2}, f_{ref})
$e_{hyp_{mt}}$	881	1350	22,340	35,213	terpa($e_{hyp_{mt}}$, e_{ref})
$e_{hyp_{slt1}}$	881*3	1350*3	61,787	97,977	terpa($e_{hyp_{slt1}}$, e_{ref})
$e_{hyp_{slt2}}$	881*3	1350*3	62,213	97,804	terpa($e_{hyp_{slt2}}$, e_{ref})
e_{ref}	881	1350	22,342	34, 880	

Table 7 Example of quintuplet with associated labels

f_{ref}	quand	notre	cerveau	chauffe
f_{hyp1}	*comme*	notre	cerveau	chauffe
labels ASR	B	G	G	G
f_{hyp2}	*qu'*	*entre*	*serbes*	*au* chauffe
labels ASR	B	B	B	B G
$e_{hyp_{mt}}$	when	our	brains	*chauffe*
labels MT	G	G	G	B
$e_{hyp_{slt1}}$	*as*	our	brains	*chauffe*
labels SLT	B	G	G	B
$e_{hyp_{slt2}}$	*between*	*serbs*	*in*	*chauffe*
labels SLT	B	B	B	B
e_{ref}	when	our	brain	heats up

- *MT* extract *good*/*bad* labels by calculating TERp-A between $e_{hyp_{mt}}$ and e_{ref},
- *SLT* extract *good*/*bad* labels by calculating TERp-A between $e_{hyp_{slt}}$ and e_{ref}.

Table 7 gives an example of the quintuplet available in our corpus. One transcript (f_{hyp1}) has 1 error while the other one (f_{hyp2}) has 4. This leads to respectively 2 B labels ($e_{hyp_{slt1}}$) and 4 B labels ($e_{hyp_{slt2}}$) in the speech translation output, while $e_{hyp_{mt}}$ has only one B label.

Tables 8 and 9 summarize the baseline ASR, MT and SLT performances obtained on our corpora, as well as the distribution of *good* (G) and *bad* (B) labels inferred for both tasks. Logically, the percentage of (B) labels increases from the MT to the SLT task under the same conditions.

5 Experiments on WCE for SLT

5.1 SLT quality assessment using only MT or ASR features

We first report in Table 11 the baseline WCE results obtained using MT or ASR features separately. In short, we evaluate the performance of 4 WCE systems for different tasks:

Table 8 MT and SLT performances on our *dev* set

Task	ASR (WER) (%)	MT (BLEU) (%)	% G (*good*)	% B (*bad*)
MT	0	49.13	76.93	23.07
SLT (ASR1)	21.86	26.73	62.03	37.97
SLT (ASR2)	16.90	28.89	63.87	36.13

Table 9 MT and SLT performances on our *tst* set

Task	ASR (WER) (%)	MT (BLEU) (%)	% G (*good*)	% B (*bad*)
MT	0	57.87	81.58	18.42
SLT ($ASR1$)	17.37	36.21	70.59	29.41
SLT ($ASR2$)	12.50	38.97	72.61	27.39

Table 10 WCE performance baseline ($\%F_G$, $\%F_B$, *%F-mes*) on *ASR1* and on *ASR2* for *tst* set (random classifier generating G or B)

	WCE for ASR			WCE for SLT		
	$\%F_G$	$\%F_B$	*%F-mes*	$\%F_G$	$\%F_B$	*%F-mes*
Task *ASR1*						
Baseline	62.99	23.11	43.05	58.27	36.70	47.49
Task *ASR2*						
Baseline	64.31	17.65	40.98	59.37	35.56	47.47

- The first and second systems (WCE for ASR/ASR feat.) use ASR features described in Sect. 3.1 with two different classifiers (CRF or Boosting).
- The third system (WCE for SLT/MT feat.) uses only the MT features described in Sect. 3.2 with the CRF classifier.
- The fourth system (WCE for SLT/ASR feat.) uses only the ASR features described in Sect. 3.1 with the CRF classifier, i.e. predicting SLT output confidence using only ASR confidence features! Word-alignment information between f_{hyp} and e_{hyp} is used to project the WCE scores coming from ASR to the SLT output.

In all experiments reported in this paper, we evaluate the performance of our classifiers by using the average between the F-measure for *good* labels and the F-measure for *bad* labels that are calculated by the common evaluation metrics: Precision, Recall and F-measure for *good*/*bad* labels. Since two ASR systems are available, *F-mes1* is obtained for SLT based on $ASR1$, whereas *F-mes2* is obtained for SLT based on $ASR2$. For the results in Table 11, the classifier is evaluated on the *tst* part of our corpus and trained on the *dev* part.

Concerning WCE for ASR, we observe that F-measure decreases when ASR WER is lower *F-mes2*<*F-mes1* while $WER_{ASR2} < WER_{ASR1}$, i.e. quality assessment in ASR seems to become harder as the ASR system improves. This could be due

Table 11 WCE performance with different feature sets for *tst* set (training is made on *dev* set)

Task	WCE for ASR	WCE for ASR	WCE for SLT	WCE for SLT
feat. type	ASR feat. $p(q\|x_f, f)$ (CRFs)	ASR feat. $p(q\|x_f, f)$ (Boosting)	MT feat. $p(q\|f, e)$	ASR feat. $p_{ASR}(q\|x_f, f)$ projected to e
F-mes1	68.71%	64.27%	64.69%*	53.85%
F-mes2	59.83%	62.61%	64.48%*	48.67%

*For MT feat, removing *OccurInGoogleTranslate* and *OccurInBingTranslate* features lead to 63.09% and 62.33% for *F-mes1* and *F-mes2*, respectively

to the fact that the ASR1 errors recovered by a bigger LM in the ASR2 system were easier to detect. Anyway, this conclusion should be considered with caution since both results (*F-mes1* and *F-mes2*) are not directly comparable because they are evaluated on different references (the proportion of *good*/*bad* labels differ as the ASR systems themselves differ). The effect of the classifier (CRF or Boosting) is not conclusive since CRF is better for *F-mes1* and worse for *F-mes2*. In any case, we decide to use CRF for all our future experiments since this is the classifier integrated in the WCE-LIG toolkit (Servan et al. 2015).

To assess WCE for SLT, the observed F-measure is better using MT features rather than ASR features, i.e. quality assessment for SLT is more dependent on MT features than ASR features. Again, F-measure decreases when ASR WER is lower (*F-mes2*<*F-mes1* while $WER_{ASR2} < WER_{ASR1}$). For MT features, removing *OccurInGoogleTranslate* and *OccurInBingTranslate* features lead to 63.09% and 62.33% for *F-mes1* and *F-mes2*, respectively. Finally, it is worth mentioning that the perfomance obtained by the classifiers in Table 11 are above the random baselines in Table 10.

In the next subsection, we investigate whether the use of both MT and ASR features improves quality assessment for SLT.

5.2 SLT quality assessment using both MT and ASR features

We report in Table 13 the WCE results for SLT obtained using both MT and ASR features. More precisely, we evaluate two different approaches (*combination* and *joint*):

- The first system (WCE for SLT/MT+ASR feat.) combines the output of two separate classifiers based on ASR and MT features. In this approach, the ASR-based confidence score of the source is projected to the target SLT output and combined with the MT-based confidence score as shown in Eq. (6) (we did not tune the α coefficient and set it *a priori* to 0.5).
- The second system (joint feat.) trains a single WCE system for SLT (evaluating $p(q|x_f, f, e)$ as in Eq. (5) using joint ASR features and MT features. All ASR features are projected to the target words using automatic word alignments. However, a problem occurs when a target word does not have any source word aligned to it. In this case, we decide to duplicate the ASR features of its previous target

Table 12 Different strategies to project ASR features to a target word when it is aligned to more than one source word

ASR feat	Joint 1	Joint 2	Joint 3
F-post	avg(F-post1, F-post2)	avg(F-post1, F-post2)	avg(F-post1, F-post2)
F-log	avg(F-log1, F-log2)	avg(F-log1, F-log2)	avg(F-log1, F-log2)
F-back	avg(F-back1, F-back2)	avg(F-back1, F-back2)	avg(F-back1, F-back2)
F-dur	max(F-dur1, F-dur2)	max(F-dur1, F-dur2)	max(F-dur1, F-dur2)
F-3g	max(F-3g1, F-3g2)	max(F-3g1, F-3g2)	max(F-3g1, F-3g2)
F-alt	max(F-alt1, F-alt2)	max(F-alt1, F-alt2)	max(F-alt1, F-alt2)
F-word	F-word1	F-word2	F-word1_F-word2
F-POS	F-POS1	F-POS2	F-POS1_F-POS2
F-context	F-context*	F-context	F-context

* It should be noted that **F-context** features are the combinations of the source word (**F-word**) and one POS of the source word (**F-POS**) before and one POS of the source word (**F-POS**) after

Table 13 WCE performance with combined (MT + ASR) or joint (MT, ASR) feature sets for *tst* set (training is made on *dev* set)

Task	WCE for SLT	WCE for SLT	WCE for SLT	WCE for SLT
feat. type	MT+ASR feat. $p_{ASR}(q\|x_f, f)^{\alpha}$ $p_{MT}(q\|e, f)^{1-\alpha}$*	Joint feat. 1 $p(q\|x_f, f, e)$	Joint feat. 2 $p(q\|x_f, f, e)$	Joint feat. 3 $p(q\|x_f, f, e)$
F-mes1	58.07%	64.90%*	64.84%	64.86%
F-mes2	53.66%	64.17%*	64.11%	63.87%

*For *Joint 1* feat, removing *OccurInGoogleTranslate* and *OccurInBingTranslate* features lead to 63.31 and 62.16% for *F-mes1* and *F-mes2*, respectively

word. Another problem occurs when a target word is aligned to more than one source word. In that case, there are several strategies we can use to infer the 9 ASR features: average or max over numerical values, selection or concatenation over symbolic values (for F-word and F-POS), etc. Three different variants of these strategies (shown in Table 12) are evaluated here.

The results in Table 13 show that joint ASR and MT features only slightly improve WCE performance: *F-mes1* is slightly better than one set-up in Table 11 (WCE for SLT/MT features only). We also observe that simple combination (MT + ASR) degrades WCE performance. This may be due to the different behaviour of the WCE_{MT} and WCE_{ASR} classifiers which makes the weighted combination ineffective. The relatively disappointing performance of our joint classifier may be due to an insufficient training set (only 2643 utterances in *dev*). Finally, removing *OccurInGoogleTranslate* and *OccurInBingTranslate* features for *Joint* lowered *F-mes* between 1 and 2%.

These observations lead us to investigate the behaviour of our WCE approaches for a large range of *good*/*bad* decision thresholds.

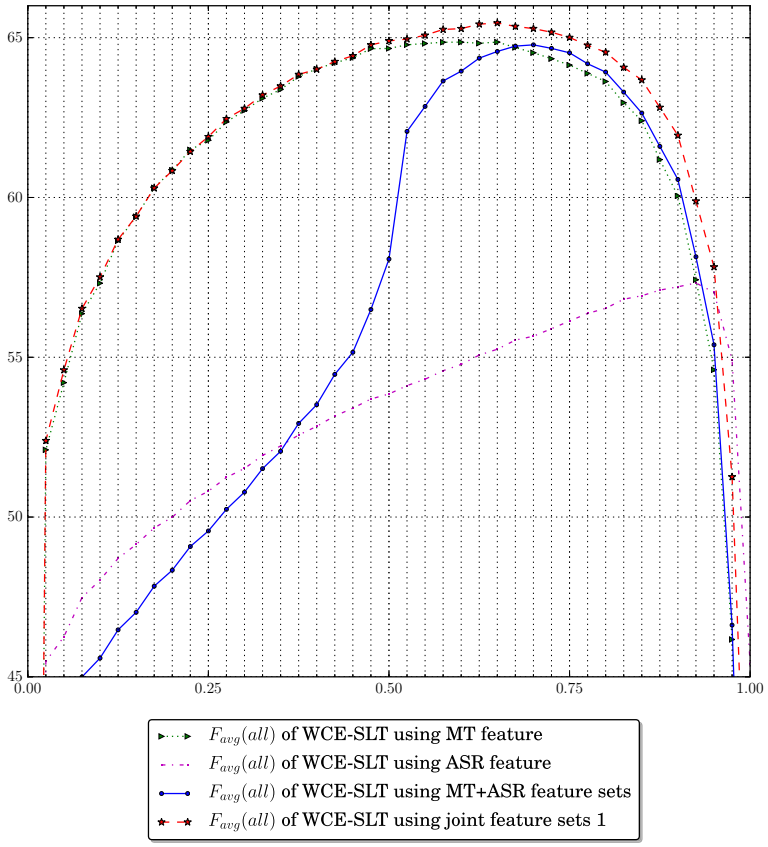

Fig. 1 Evolution of system performance (y-axis – *F-mes1* – ASR1) for *tst* corpus (4050 utt) along decision threshold variation (x-axis). Training is made on *dev* corpus (2643 utt)

While the previous tables provided WCE performance for a single point of interest (*good*/*bad* decision threshold set to 0.5), the curves in Figures 1 and 2 show the full picture of our WCE systems (for SLT) using speech-transcription systems $ASR1$ and $ASR2$, respectively. We observe that the classifier based on ASR features has a very different behaviour than the classifier based on MT features which explains why their simple combination (MT + ASR) does not work very well for the default decision threshold (0.5). However, for thresholds above 0.75, the use of joint ASR and MT features is slightly beneficial compared to MT features only. This is interesting because higher thresholds improve the F-measure on *bad* labels and so improve error detection. Both curves are similar whatever the ASR system used. These results suggest that with enough development data for appropriate threshold tuning (which we do not have for this very new task), the use of both ASR and MT features should improve error detection in speech translation (blue and red curves are above the green curve for higher decision thresholds).[12] Although not reported here, we also analyzed the

[12] Corresponding to optimization of the F-measure on *bad* labels (errors).

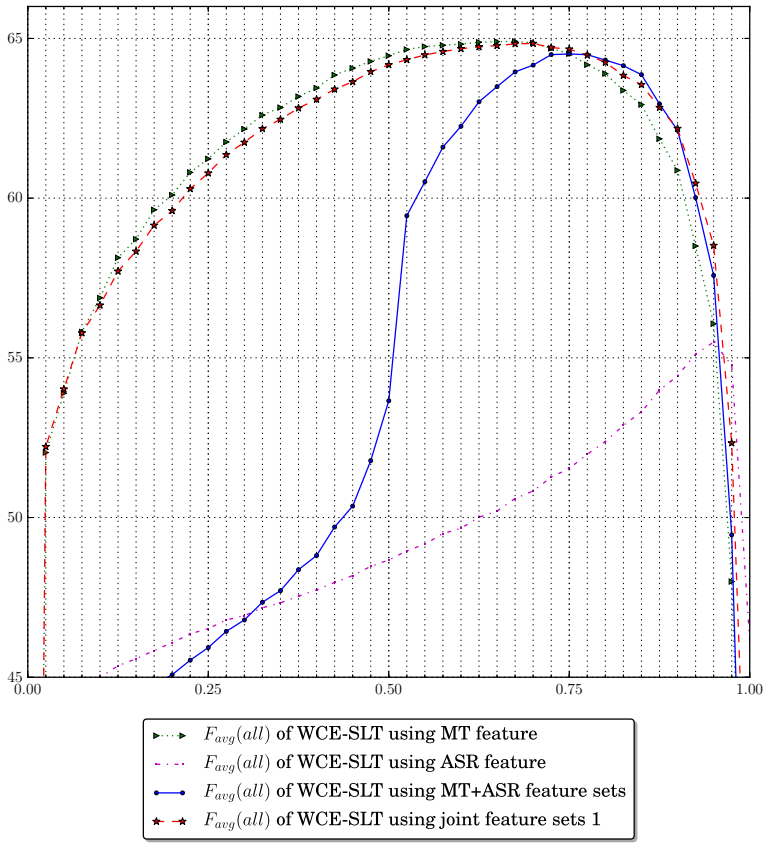

Fig. 2 Evolution of system performance (y-axis – *F-mes2* – ASR2) for *tst* corpus (4050 utt) along decision-threshold variation (x-axis). Training is made on *dev* corpus (2643 utt)

F-measure curves for *bad* and *good* labels separately: if we consider, for instance, the $ASR1$ system, for decision threshold equal to 0.75, the F-measure on *bad* labels is equivalent (52%) for 3 systems (*Joint, MT + ASR* and *MT*), while the F-measure on *good* labels is 76% when using *MT* features only, 78% when using *Joint* features and 77% when using *MT + ASR* features. In other words, for a fixed performance on *bad* labels, the F-measure on *good* labels is improved using all information available (ASR and MT features). Finally, if we focus on *Joint* versus *MT + ASR*, we notice that the range of the threshold where performance is stable is larger for *Joint* than for *MT + ASR*.

6 Feature selection

In this section, we try to better understand the contribution of each (ASR or MT) feature by applying feature selection on our joint WCE classifier. In these experiments, we decide to keep *OccurInGoogleTranslate* and *OccurInBingTranslate* features.

We choose the Sequential Backward Selection (SBS) algorithm (Aha and Bankert 1996) which is a top-down algorithm starting from a feature set noted Y_k (which denotes the set of all features) and sequentially removing the most irrelevant one (x) that maximizes the Mean F-Measure, $MF(Y_k - x)$. In our work, we examine until the set Y_k contains only one remaining feature. Algorithm 1 summarizes the whole process.

Algorithm 1 Sequential Backward Selection (SBS) algorithm for feature selection. Y_k denotes the set of all features and x is the feature removed at each step of the algorithm.

```
while size of Y_k > 0 do
    maxval = 0
    for x ∈ Y_k do
        if maxval < MF(Y_k − x) then
            maxval ← MF(Y_k − x)
            worstfeat ← x
        end if
    end for
    remove worstfeat from Y_k
end while
```

The results of the SBS algorithm can be found in Table 14 which ranks all joint features used in WCE for SLT by order of importance after applying the algorithm on *dev*. We can see that the SBS algorithm is not very stable and is clearly influenced by the ASR system ($ASR1$ or $ASR2$) considered in SLT. In any case, if we focus on the 10-best features in both cases, we find that the most relevant ones are:

- *Alignment Features* (source and target collocation features),
- *Occur in Google Translate* and *Occur in Bing Translate* (diagnostic from other MT systems),
- *Longest Source N-gram Length, Target Backoff Behaviour* (source or target *N*-gram features),
- *Word Posterior Probability Max (WPP Max)* (graph topology feature),

We also observe that the most relevant ASR features (in Table 14) are *F-back*, *F-3g* and *F-context* (linguistic and context features) whereas ASR lexical, acoustic and graph-based features are among the worst (*F-POS, F-dur* and *F-post*). Accordingly, in our experimental setting, it seems that MT features are more influential than ASR features. Interestingly, "source and target collocation features" (Alignment Features) and "Occur in Bing Translate" are the most prominent features (rank 1 and rank 2, respectively) when applied to *dev* corpus for both *ASR1* and *ASR2*. Besides, the graph-topology feature extracted from a confusion network *WPP Max* outperforms the others such as *Nodes* and *WPP Min*. Nevertheless, two other features including *WPP Exact* and *WPP any* are proven to be weak in accordance with their bottom-most positions against the two above systems, whereas we were expecting to see them among the top features (as shown in Luong et al. (2015) where *WPP Any* is among the best features for WCE in MT).

Table 14 Rank of each feature according to the sequential backward selection algorithm on the WCE for SLT task using Joint (ASR,MT) features

Rank *ASR1*	Rank *ASR2*	Feature	Rank *ASR1*	Rank *ASR2*	Feature
1	1	Alignment Features	18	20	Unknown Stem
2	2	Occur in Bing Translate	19	29	Number of Word Occurrences
3	4	Longest Source *N*-gram Length	20	28	Polysemy Count - Target
4	3	WPP Max	21	19	**F-dur**
5	6	Occur in Google Translate	22	12	Punctuation
6	24	**F-back**	23	21	Constituent Label
7	11	**F-context**	24	25	**F-word**
8	27	**F-alt**	25	23	Longest Target *N*-gram Length
9	7	Target Backoff Behaviour	26	10	POS Context Alignment
10	5	Word Context Alignment	27	26	WPP Exact
11	30	Stem Context Alignment	28	18	WPP Any
12	31	Numeric	29	22	Proper Name
13	13	Distance to Root	30	8	Number of Stem Occurrences
14	9	**F-3g**	31	16	**F-POS**
15	17	Stop Word	32	33	**F-post**
16	15	Nodes	33	32	**F-log**
17	14	WPP Min			

Feature selection is applied to the *dev* corpus for both *ASR1* and *ASR2*. ASR features are in bold

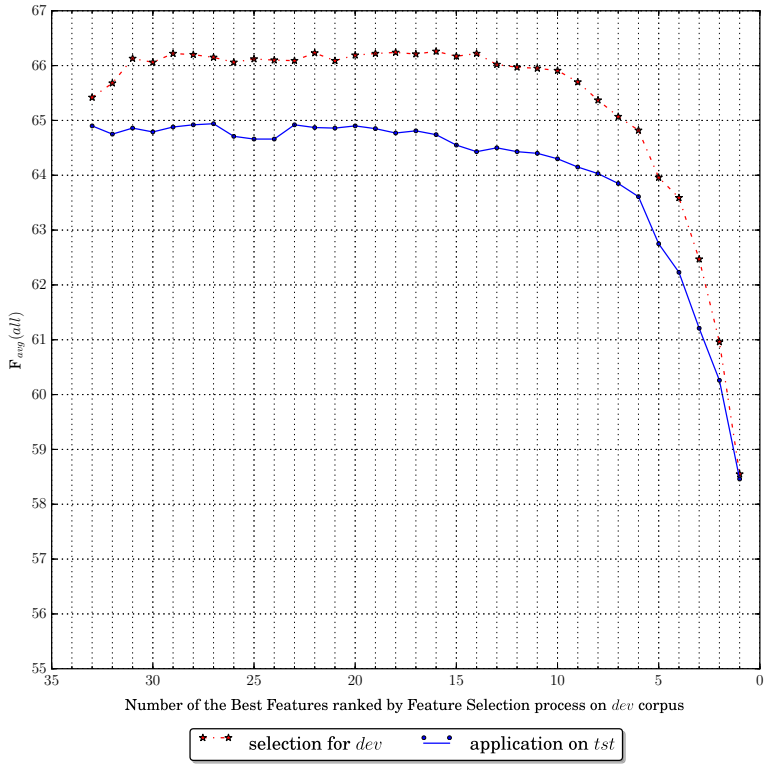

Fig. 3 Evolution of WCE performance for *dev* (features selected) and *tst* corpora when feature selection using SBS algorithm is made on *dev* ($ASR1$ system only; the same shape is observed for $ASR2$)

Figure 3 presents the evolution of WCE performance for *dev* and *tst* corpora when feature selection using the SBS algorithm is made on *dev*, for the $ASR1$ system; the same shape is observed for the $ASR2$ system. In other words, for this figure, we apply our SBS algorithm on *dev* which means that feature selection is done on *dev* with classifiers trained on *tst*. After that, the best feature subsets (using 33, 32, 31 until 1 feature only) are applied to the *tst* corpus (with classifiers trained on *dev*).[13]

In the figure, we observe that half of the features only contribute to the WCE process since the best performance is observed with only 15 to 25 features. We also see that optimal WCE performance is not necessarily obtained with the full feature set, as it can be obtained with a subset of it.

[13] In principle, three data sets would have been needed to (a) train classifiers, (b) apply feature selection, (c) evaluate WCE performance. Since we only have a *dev* and a *tst* set, we found this procedure acceptable.

7 Disentangling ASR and MT errors

In the previous section, we only extract *good*/*bad* labels from the SLT output while it might be interesting to move from a 2-class problem to a 3-class problem in order to label our SLT hypotheses with one of the 3 following labels: *good* (*G*), *asr-error* (*B_ASR*) and *mt-error* (*B_MT*). Before training automatic systems for error detection, we need to set such 3-class labels for our *dev* and *test* corpora. For that, in the next subsections we propose two slightly different methods to extract them. The first one is based on the word alignments from SLT to MT, and the second is based on a subtraction between SLT and MT errors.

7.1 Method 1: using word alignments between MT and SLT

In MT, the fertility of a source word denotes how many output words it translates as. If we transpose this definition to our disentangling problem, then *fertility of an MT error* denotes how many erroneous words—in the SLT output—it is aligned to. From this simple definition, we derive our first way (*Method 1*) of generating 3-class annotations.

Let $\hat{e}_{slt} = (e_1, e_2, \ldots, e_n)$ be the set of SLT hypotheses ($e_{hyp_{slt}}$); e_{k_j} denotes the jth word in the sentence e_k, where $1 \leq k \leq n$.

Let $\hat{e}_{mt} = (e'_1, e'_2, \ldots, e'_n)$ be the set of MT hypotheses ($e_{hyp_{mt}}$); e'_{k_i} denotes the ith word in the sentence e'_k, where $1 \leq k \leq n$.

Let $L = (l_1, l_2, \ldots, l_n)$ be the set of word alignments from sentences in $e_{hyp_{slt}}$ to related sentences in $e_{hyp_{mt}}$, where l_k contains the word alignments from sentence e_k to the relevant sentence e'_k, $1 \leq k \leq n$; (e_{k_j}, e'_{k_i}) = *True*, if there is one word alignment between e_{k_j} and e'_{k_i}; (e_{k_j}, e'_{k_i}) = *False*, otherwise.

Our algorithm for *Method 1* is defined as *Algorithm 2*. This method relies on word alignments and uses MT labels. We also propose a simpler method in the next section.

Algorithm 2 *Method 1*: Using word alignments between MT and SLT

```
list_labels_result ← empty_list
for each sentence e_k ∈ ê_slt do
    list_labels_sent ← empty_list
    for j ← 1 to NumberOfWords(e_k) do
        if label(e_kj) = 'G' then
            add 'G' to list_labels_sent
        else if Existed Word Alignment (e_kj, e'_ki) and label(e'_ki)='B' then
            add 'B_MT' to list_labels_sent
        else
            add 'B_ASR' to list_labels_sent
        end if
    end for
    add list_labels_sent to list_labels_result
end for
```

7.2 Method 2: subtraction between SLT and MT errors

Our second way to extract 3-class labels (*Method 2*) focuses on the differences between the SLT hypothesis ($e_{hyp_{slt}}$) and the MT hypothesis ($e_{hyp_{mt}}$). We call it *subtraction between SLT and MT errors* because we simply consider that errors present in SLT and not present in MT are due to ASR. This method has one main difference with the previous one, namely that it does not rely on the extracted labels for MT.

Our intuition is that the number of *mt-error*s estimated will be slightly lower than for *Method 1* since we first estimate the number of *asr-error*s and the rest is considered—by default—as *mt-error*s.

We use the same notations as *Method 1*, except that $L = (l_1, l_2, \ldots, l_n)$ is the set of alignments through edit distance between $e_{hyp_{slt}}$ and $e_{hyp_{mt}}$, where l_{k_i} corresponds to "Insertion", "Substitution", "Deletion" or "Exact". Our algorithm for *Method 2* is defined in *Algorithm 3*.

Algorithm 3 *Method 2*: Subtraction between SLT and MT errors

```
list_labels_result ← empty_list
for each sentence e_k ∈ ê_slt do
    list_labels_sent ← empty_list
    for j ← 1 to NumberOfWords(e_k) do
        if label(e_kj) = 'G' then
            add 'G' to list_labels_sent
        else if NameOfWordAlignment(l_ki) is 'Insertion' OR 'Substitution' then
            add 'B_ASR' to list_labels_sent
        else
            add 'B_MT' to list_labels_sent
        end if
    end for
    add list_labels_sent to list_labels_result
end for
```

7.3 Example with 3-label setting

Table 15 gives the edit distance between an SLT and an MT hypothesis, while Table 16 shows how *Method 1* and *Method 2* set 3-class labels to the SLT hypothesis. One transcript (f_{hyp}) has 1 error. This drives 3 B labels on SLT output ($e_{hyp_{slt}}$), while $e_{hyp_{mt}}$ has only 2 B labels. As can be seen for *Method 1* and *Method 2*, we respectively have (1 B_ASR, 2 B_MT) and (2 B_ASR, 1 B_MT).

Table 15 Example of edit distance between SLT and MT

$e_{hyp_{slt}}$	surgeons	in	los	angeles	it	is	said
$e_{hyp_{mt}}$	surgeons	in	los	angeles	**	have	said
	E	E	E	E	I	S	E

Table 16 Example of quintuplet with 2 and 3-label

f_{hyp}	les chirurgiens	de	los	angeles	on		dit
labels ASR	G G	G	G	G	B		G
$e_{hyp_{mt}}$	surgeons	in	los	angeles		have	said
labels MT	G	B	G	G		B	G
$e_{hyp_{slt}}$	surgeons	in	los	angeles	it	is	said
labels SLT (2-label)	G	B	G	G	B	B	G
labels SLT (*Method 1*)	G	B_MT	G	G	B_ASR	B_MT	G
labels SLT (*Method 2*)	G	B_MT	G	G	B_ASR	B_ASR	G
e_{ref}	the surgeons	of	los	angeles			said

Table 17 Statistics with 3-label setting for *ASR1*

Task	*dev* set			*tst* set		
	%G	*%B_ASR*	*%B_MT*	%G	%B_ASR	%B_MT
label/m1:Method 1	62.03	19.09	18.89	70.59	14.50	14.91
label/m2:Method 2	62.03	22.49	15.49	70.59	16.62	12.79
label/same(m1, m2)	62.03	18.09	14.49	70.59	13.58	11.88
label/diff(m1, m2)	0	1.00	4.40	0	0.92	3.03

Table 18 Statistics with 3-label setting for *ASR2*

Task	*dev* set			*tst* set		
	%G	*%B_ASR*	*%B_MT*	%G	%B_ASR	%B_MT
label/m1:Method 1	63.87	16.89	19.23	72.61	11.92	15.47
label/m2:Method 2	63.87	19.78	16.34	72.61	13.58	13.81
label/same(m1, m2)	63.87	16.05	15.50	72.61	11.12	13.01
label/diff(m1, m2)	0	0.84	3.73	0	0.80	2.46

These differences might be related to the word alignments from the SLT hypothesis to the relevant MT hypothesis. As Table 16 presents, “is” (SLT hypothesis) is aligned to “have” (MT hypothesis) and “have” (MT hypothesis) is labeled with “B”. It can be assumed, therefore, that “is” (SLT hypothesis) should be annotated with word-level labels by B_MT according to *Method 1*. However, using *Method 2*, “is” (SLT hypothesis) could be labeled B_ASR because the type of word alignment between “is” (SLT hypothesis) and “have” (MT hypothesis) is substitution (S), as shown in Table 15.

7.4 Statistics with 3-label setting on the whole corpus

Tables 17 and 18 present the summary statistics for the distribution of *good* (G), *asr-error* (B_ASR) and *mt-error* (B_MT) labels obtained with both label-extraction methods. We see that both methods give similar statistics but slightly different rates of B_ASR and B_MT.

Comparing Tables 17 and 18, it is interesting to note that while the ASR system improves from *ASR1* to *ASR2*, the rate of B_ASR labels logically decreases by more than 2 points, while the rate of B_MT remains almost stable (less than 1 point difference) which makes sense since the MT system is the same in both Tables 17 and 18. These statistics show that the intersection between both methods is probably a good estimation of disentangling ASR and MT errors in SLT.

7.5 Qualitative analysis of SLT errors

Our new 3-label setting procedure allows us to analyze the behaviour of our SLT system. We omit examples here, but they are made available as supplementary material

Table 19 Error-detection performance (2 vs. 3-labels) on SLT output for the tst set

Error detection	2-class			3-class	
	ASR1	*ASR2*		*ASR1*	*ASR2*
F_G	81.79	83.17	F_G	85.00	85.00
F_B	48.00	45.17	F_{B_ASR}	44.00	42.00
			F_{B_MT}	14.00	15.00
F_{avg}	64.90	64.17	F_{avg}	47.67	47.33

Training is performed on the *dev* set

to this paper on a Web link.[14] Nonetheless, we can observe sentences with few ASR and MT errors leading to many SLT errors. Indeed, this is a good way of detecting flaws in the SLT pipeline such as bad post-processing of the SLT output (numerical or text dates, for instance). In contrast, there are cases where many ASR errors lead to few SLT errors (ASR errors with few consequences such as morphological substitutions (for instance in French: de/des, déficit/déficits, budgétaire/budgétaires). Finally, some ASR errors have different consequences on SLT quality (on a sample sentence, 2 ASR errors in Systems 1 and 2 lead to 14 and 9 SLT errors, respectively).

7.6 Experiments on 3-class error detection

We report in Table 19 our first attempt to build an error-detection system in SLT as a 3-class problem (*joint* approach only). We conducted our experiment by training and evaluating the model on *Intersection*(m1, m2) which corresponds to high confidence in the labels.[15] In addition to giving a better informed error detection (B_{ASR} and B_{MT} instead of B), we note that 3-class error detection leads to overall similar results if we backoff to *good/bad* decision (F_{avg} becomes 62.5 on ASR1 and 61.00 on ASR2 in that case).

8 Conclusion

8.1 Main contributions

In this paper, we introduced a new quality-assessment task: word confidence estimation (WCE) for spoken language translation (SLT). A specific corpus distributed to the research community was built for this purpose. We formalized WCE for SLT and proposed several approaches based on several types of features: MT-based features, ASR-based features, as well as combined or joint features using both ASR and MT information. The proposal of a unique *joint* classifier based on different feature types

[14] http://tienhuong.weebly.com/examples-for-the-paper.html.

[15] However, we observed that the use of different label sets (*Method* 1, *Method* 2, *Intersection(Method* 1, *Method* 2)) does not have a strong influence on the results, so we omit these results here.

(ASR and MT features) allowed us to operate feature selection and analyze which features (from ASR or MT) are the most efficient for quality assessment in speech translation. Our experiments have shown that MT features remain the most influential, while ASR features can bring interesting complementary information. For the purpose of reproducible research, our toolkit has been made available on a *GitHub* repository under the licence GPL V3. We hope that the availability of our corpus and toolkit could lead, in the near future, to a new shared task dedicated to quality estimation for speech translation. Such a shared task could be proposed in avenues such as IWSLT or WMT for instance. Towards the end of the paper, we proposed to disentangle ASR and MT errors and recast WCE as a 3-label setting problem.

8.2 Perspectives

A direct application of this work is the use of WCE labels to re-decode speech-translation graphs and (hopefully) improve speech-translation performance. Preliminary results have already been obtained and published by the authors of this paper (Besacier et al. 2015). The main idea is to carry a second speech translation pass by considering every word and its quality-assessment label, as shown in Eq. (4).

In addition to re-decoding SLT graphs, our quality-assessment system can be used in interactive speech-translation scenarios such as news or lecture subtitling, to improve human translator productivity by giving them feedback on automatic transcription and translation quality. Another application would be the adaptation of our WCE system to interactive speech-to-speech translation scenarios where feedback on transcription and translation modules is needed to improve communication. Finally, in this paper, engineered features were used for WCE; a natural perspective is also to learn the WCE features as it is now possible with deep neural networks, for instance.

References

Aha DW, Bankert RL (1996) A comparative evaluation of sequential feature selection algorithms. In: Fisher D, Lenz HJ (eds) Learning from data: artificial intelligence and statistics V. Springer, New York, pp 199–206

Asadi A, Schwartz R, Makhoul J (1990) Automatic detection of new words in a large vocabulary continuous speech recognition system. In: Proceedings of the international conference on acoustics, speech and signal processing, Albuquerque, New Mexico, USA, pp 263–265

Bach N, Huang F, Al-Onaizan Y (2011) Goodness: a method for measuring machine translation confidence. In: Proceedings of the 49th annual meeting of the association for computational linguistics, Portland, Oregon, pp 211–219

Besacier L, Lecouteux B, Luong NQ, Hour K, Hadjsalah M (2014) Word confidence estimation for speech translation. In: Proceedings of the international workshop on spoken language translation (IWSLT), Lake Tahoe, CA, USA, pp 169–175

Besacier L, Lecouteux B, Luong NQ, Le NT (2015) Spoken language translation graphs re-decoding using automatic quality assessment. In: Proceedings of IEEE automatic speech recognition and understanding workshop (ASRU 2015), Scotsdale, Arizona, United States, p 8

Biçici E (2013) Referential translation machines for quality estimation. In: Proceedings of the eighth workshop on statistical machine translation, Sofia, Bulgaria, pp 343–351

de Souza JGC, Zamani H, Negri M, Turchi M, Falavigna D (2015) Multitask learning for adaptive quality estimation of automatically transcribed utterances. In: Proceedings of the 2015 conference of the North

American chapter of the association for computational linguistics: human language technologies. Denver, Colorado, USA, pp 714–724

Fayolle J, Moreau F, Raymond C, Gravier G, Gros P (2010) CRF-based combination of contextual features to improve a posteriori word-level confidence measures. In: Proceedings of the 11th annual conference of the international speech communication association, Makuhari, Chiba, Japan, pp 1942–1945

Federico M, Cettolo M, Bentivogli L, Paul M, Stüker S (2012) Overview of the IWSLT 2012 evaluation campaign. In: Proceedings of the 9th international workshop on spoken language translation (IWSLT), Hong Kong, China, pp 12–33

Galliano S, Geoffrois E, Gravier G, Bonastre JF, Mostefa D, Choukri K (2006) Corpus description of the ESTER evaluation campaign for the rich transcription of French broadcast news. In: In Proceedings of the 5th international conference on language resources and evaluation (LREC 2006), Genoa, Italy, pp 315–320

Han ALF, Lu Y, Wong DF, Chao LS, He L, Xing J (2013) Quality estimation for machine translation using the joint method of evaluation criteria and statistical modeling. In: Proceedings of the eighth workshop on statistical machine translation, Sofia, Bulgaria, pp 365–372

Jalalvand S, Negri M, Turchi M, C de Souza JG, Daniele F, Qwaider MRH (2016) Transcrater: a tool for automatic speech recognition quality estimation. In: Proceedings of ACL-2016 system demonstrations, Berlin, Germany, pp 43–48

Kemp T, Schaaf T (1997) Estimating confidence using word lattices. In: Proceedings of the European conference on speech communication technology, Rhodes, Greece, pp 827–830

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, Herbst E (2007) Moses: open source toolkit for statistical machine translation. In: Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions, Prague, Czech Republic, pp 177–180

Lafferty J, McCallum A, Pereira F (2001) Conditional random fields: Probabilistic models for segmenting et labeling sequence data. In: Proceedings of the eighteenth international conference on machine learning (ICML 2001), Williamstown, MA, USA, pp 282–289

Langlois D, Raybaud S, Smaïli K (2012) Loria system for the WMT12 quality estimation shared task. In: Proceedings of the seventh workshop on statistical machine translation, Montreal, Quebec, Canada, pp 114–119

Laurent A, Camelin N, Raymond C (2014) Boosting bonsai trees for efficient features combination: application to speaker role identification. In: Proceedings of the 15th annual conference of the international speech communication association (INTERSPEECH 2014), Singapore, pp 76–80

Lavergne T, Cappé O, Yvon F (2010) Practical very large scale CRFs. In: Proceedings of the 48th annual meeting of the association for computational linguistics, Uppsala, Sweden, pp 504–513

Lecouteux B, Linarès G, Favre B (2009) Combined low level and high level features for out-of-vocabulary word detection. In: Proceedings of the 10th annual conference of the international speech communication association (INTERSPEECH 2009), Brighton, UK, pp 1187–1190

Levenshtein V (1966) Binary codes capable of correcting deletions, insertions, and reversals. Soviet Phys Doklady 10:707–710

Logacheva V, Hokamp C, Specia L (2016) MARMOT: a toolkit for translation quality estimation at the word level. In: Proceedings of the tenth international conference on language resources and evaluation (LREC 2016), Portorož, Slovenia, p 4

Luong NQ, Besacier L, Lecouteux B (2013a) Word confidence estimation and its integration in sentence quality estimation for machine translation. In: Proceedings of the fifth international conference on knowledge and systems engineering (KSE 2013), Hanoi, Vietnam, p 12

Luong NQ, Besacier L, Lecouteux B (2014a) LIG system for word level QE task at WMT14. In: Proceedings of the ninth workshop on statistical machine translation, Baltimore, Maryland, USA, pp 335–341

Luong NQ, Besacier L, Lecouteux B (2014b) Word confidence estimation for SMT N-best list re-ranking. In: Proceedings of the EACL 2014 workshop on humans and computer-assisted translation, Gothenburg, Sweden, pp 1–9

Luong NQ, Lecouteux B, Besacier L (2013b) LIG system for WMT13 QE task: investigating the usefulness of features in word confidence estimation for MT. In: Proceedings of the eighth workshop on statistical machine translation, association for computational linguistics, Sofia, Bulgaria, pp 396–391

Luong NQ, Besacier L, Lecouteux B (2015) Towards accurate predictors of word quality for machine translation: lessons learned on french-english and english-spanish systems. Data Knowl Eng 96:32–42

Ng RWM, Doulaty M, Doddipatla R, Aziz W, Shah K, Saz O, Hasan M, AlHarbi G, Specia L, Hain T (2014) The USFD spoken language translation system for IWSLT 2014. In: Proceedings of the international workshop on spoken language translation (IWSLT), Lake Tahoe, CA, USA, pp 86–91

Ng RWM, Shah K, Aziz W, Specia L, Hain T (2015) Quality estimation for asr k-best list rescoring in spoken language translation. In: 2015 IEEE international conference on acoustics, speech and signal processing, ICASSP 2015. Brisbane, Queensland, Australia, pp 5226–5230

Ng RWM, Shah K, Specia L, Hain T (2016) Groupwise learning for ASR k-best list reranking in spoken language translation. In: Proceedings of the 41st international conference on acoustics, speech and signal processing (ICASSP 2016), Shanghai, China, pp 6120–6124

Potet M, Besacier L, Blanchon H (2010) The LIG machine translation system for WMT 2010. In: Proceedings of the joint fifth workshop on statistical machine translation and MetricsMATR, Uppsala, Sweden, pp 161–166

Potet M, Esperança-Rodier E, Besacier L, Blanchon H (2012) Collection of a large database of french-english SMT output corrections. In: Proceedings of the eighth international conference on language resources and evaluation (LREC-2012), Istanbul, Turkey, pp 4043–4048

Povey D, Ghoshal A, Boulianne G, Burget L, Glembek O, Goel N, Hannemann M, Motlicek P, Qian Y, Schwarz P, Silovsky J, Stemmer G, Vesely K (2011) The Kaldi speech recognition toolkit. In: IEEE 2011 workshop on automatic speech recognition and understanding, Waikoloa, Hawaii, US, p 4

Servan C, Le NT, Luong NQ, Lecouteux B, Besacier L (2015) An open source toolkit for word-level confidence estimation in machine translation. In: Proceedings of the 12th international workshop on spoken language translation (IWSLT'15), Da Nang, Vietnam, pp 196–203

Snover M, Madnani N, Dorr B, Schwartz R (2009) Fluency, adequacy, or HTER? Exploring different human judgments with a tunable MT metric. In: Proceedings of the fourth workshop on statistical machine translation, Athens, Greece, pp 259–268

Specia L, Paetzold G, Scarton C (2015) Multi-level translation quality prediction with QuEst++. In: Proceedings of ACL-IJCNLP 2015 system demonstrations, Beijing, China, pp 115–120

Young SR (1994) Recognition confidence measures: detection of misrecognitions and out-of-vocabulary words. In: Proceedings of the international conference on acoustics, speech and signal processing, Adelaide, South Australia, pp 21–24

Zamani H, de Souza JG, Negri M, Turchi M, Falavigna D (2015) Reference-free and confidence-independent binary quality estimation for automatic speech recognition. In: Proceedings of the second Italian conference on computational linguistics (CLiC-it), Trento, Italy, pp 280–285