Disentangling ASR and MT Errors in Speech Translation

Ngoc-Tien Le ngoc-tien.le@univ-grenoble-alpes.fr
Benjamin Lecouteux benjamin.lecouteux@univ-grenoble-alpes.fr
Laurent Besacier laurent.besacier@univ-grenoble-alpes.fr
University Grenoble Alpes, CNRS, Grenoble INP, LIG, F-38000 Grenoble, France

Abstract

The main aim of this paper is to investigate automatic quality assessment for spoken language translation (SLT). More precisely, we investigate SLT errors that can be due to transcription (ASR) or to translation (MT) modules. This paper investigates automatic detection of SLT errors using a single classifier based on joint ASR and MT features. We evaluate both 2-class (*good*/*bad*) and 3-class ($good$/bad_{ASR}/bad_{MT}) labeling tasks. The 3-class problem necessitates to disentangle ASR and MT errors in the speech translation output and we propose two label extraction methods for this non trivial step. This enables - as a by-product - qualitative analysis on the SLT errors and their origin (are they due to transcription or to translation step?) on our large in-house corpus for French-to-English speech translation.

Index Terms: Spoken Language Translation, Automatic Speech Recognition, Confidence Estimation, Quality Estimation, ASR and MT errors detection.

1 Introduction

This paper addresses a relatively new quality assessment task: error detection in spoken language translation (SLT) using both automatic speech recognition (ASR) features and machine translation (MT) features. To our knowledge, the first attempts to design error detection for speech translation, using both ASR and MT features, are our own work (Besacier et al., 2014, 2015) which is further extended in this paper submission.

Contributions (1) This paper extends previous work (Besacier et al., 2014, 2015) in 2-class ($good$/bad) error detection in SLT using a single classifier based on joint ASR and MT features (2) in order to disentangle ASR and MT errors in SLT, we extend error detection to a 3-class problem ($good$/bad_{ASR}/bad_{MT}) where we try to find the source of the SLT errors (3) two methods are compared for setting such 3-class labels on our corpus and a first attempt to automatically detect errors and their origin in a SLT output is presented at the end of this paper.

Outline The outline of this paper goes simply as follows: Section 2 formalizes error detection in SLT and presents our experimental setup. Section 3 proposes two methods to disentangle ASR and MT errors in SLT output and presents statistics on a large French-English corpus. Section 4 presents our 2-class and 3-class error detection results while section 5 concludes this work and gives some perspectives.

2 Automatic Error Detection in Speech Translation

2.1 Formalization

A quality estimation (or error detection) component in speech translation solves the equation:

$$\hat{q} = \underset{q}{\operatorname{argmax}}\{p_{SLT}(q|x_f, f, \hat{e})\} \tag{1}$$

where x_f is the given signal in the source language; $\hat{e}$ [1] $= (e_1, e_2, ..., e_N)$ is the most probable target language sequence from the spoken language translation (SLT) process; $f = (f_1, f_2, ..., f_M)$ is the transcription of x_f; $q = (q_1, q_2, ..., q_N)$ is a sequence of error labels on the target language and $q_i \in \{good, bad\}$ [2]. This is a sequence labeling task that can be solved with several machine learning techniques such as Conditional Random Fields (CRF) (Lafferty et al., 2001). However, for that, we need a large amount of training data for which a quadruplet (x_f, f, e, q) is available.

As it is much easier to obtain data containing either the triplet (x_f, f, q) (ASR output + manual references and error labels inferred from WER) or the triplet (f, e, q) (MT output + manual post-editions and error labels inferred using tools such as TERp-A (Snover et al., 2008)) we can also recast error detection with the following equation:

$$\hat{q} = \underset{q}{\operatorname{argmax}}\{p_{ASR}(q|x_f, f)^{\alpha} * p_{MT}(q|e, f)^{1-\alpha}\} \tag{2}$$

where α is a weight giving more or less importance to error detector on transcription compared to error detector on translation.

2.2 Dataset, ASR and MT Modules

2.2.1 Dataset

In this paper, we use our in-house corpus made available on a *github* repository [3] for reproductibility. 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 *dev* set and *tst* set, respectively. For each speech utterance, a quintuplet containing: ASR output (f_{hyp}), verbatim transcript (f_{ref}), text translation output ($e_{hyp_{mt}}$), speech translation output ($e_{hyp_{slt}}$) and post-edition of translation (e_{ref}) is available. The total length of the union of *dev* and *tst* is 16h52 (42 speakers - 5h51 for *dev* and 11h01 for *tst*).

2.2.2 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, REPERE, ETAPE and BREF120) representing more than 600 hours of french transcribed speech. We use two 3-gram language models trained on French ESTER corpus (Galliano et al., 2006) as well as on French Gigaword (vocabulary size are respectively 62k and 95k). ASR systems LM weight parameters are tuned through WER on *dev* corpus. *Table 1* presents the performances obtained by both ASR systems.

2.2.3 SMT System

We used *moses* phrase-based translation toolkit (Koehn et al., 2007) to translate French ASR into English (e_{hyp}). This medium-size system was trained using a subset of data provi-

[1] written simply e for convenience in any other equations

[2] at this point q_i takes two values (G/B) but will evolve to 3 labels later on in section 3

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

ded for IWSLT 2012 evaluation (Federico et al., 2012): Europarl, Ted and News-Commentary corpora. The total amount is 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)).

2.3 Obtaining Error Labels for SLT

After building an ASR system, we have a new element of our desired quintuplet: the ASR output f_{hyp}. It is the noisy version of our already available verbatim transcripts called f_{ref}. This ASR output (f_{hyp}) is then translated by the SMT system (Potet et al., 2010) already mentioned in subsection 2.2.3. This new output translation is called $e_{hyp_{slt}}$ and it is a degraded version of $e_{hyp_{mt}}$ (translation of f_{ref}). To infer the quality (G, B) labels of our speech translation output $e_{hyp_{slt}}$, we use TERp-A toolkit (Snover et al., 2008) between $e_{hyp_{slt}}$ and e_{ref} (more details can be found in our former paper (Besacier et al., 2015)). Table 1 summarizes 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 MT to SLT task in the same conditions and it decreases when ASR system improves.

Task	*ASR (WER)*		*MT (BLEU)*		*% G (good))*		*% B (bad)*	
	dev set	*tst* set	*dev* set	*tst* set	*dev* set	*tst* set	*dev* set	*tst* set
MT			49.13%	57.87%	76.93%	81.58%	23.07%	18.42%
SLT (ASR1)	21.86%	17.37%	26.73%	36.21%	62.03%	70.59%	37.97%	29.41%
SLT (ASR2)	16.90%	12.50%	28.89%	38.97%	63.87%	72.61%	36.13%	27.39%

Table 1. ASR, MT and SLT performances on our *dev* set and *tst* set.

3 Disentangling ASR and MT Errors

In 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 on our dev and $test$ corpora. For that, we propose, in the next sub-sections, two slightly different methods to extract them. The first one is based on word alignments between SLT and MT and the second one is based on a simpler SLT-MT error subtraction.

3.1 Method 1 - Word Alignments between MT and SLT

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

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

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

Let $L = (l_1, l_2, \ldots, l_n)$: the set of the 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 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 1*. This method relies on word alignments and uses MT labels. We also propose a simpler method in the next section.

Algorithm 1 *Method 1* - Using word alignments between MT and SLT

```
$list\_labels\_result \leftarrow empty\_list$
for each sentence $e_k \in \hat{e}_{slt}$ do
    $list\_labels\_sent \leftarrow empty\_list$
    for $j \leftarrow 1$ to $NumberOfWords(e_k)$ do
        if $label(e_{k_j}) =$ 'G' then
            add 'G' to $list\_labels\_sent$
        else if Existed Word Alignment $(e_{k_j}, e'_{k_i})$ and $label(e'_{k_i})$='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
```

3.2 Method 2 - Subtraction between SLT and MT Errors

Our second way to extract 3-class labels (*Method 2*) focuses on the differences between SLT hypothesis ($e_{hyp_{slt}}$) and 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 a main difference with the previous one: 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.

With the same notations of *Method 1*, but highlighting 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 as follows.

3.3 Example with 3-label Setting

Table 2 gives the edit distance between a SLT and MT hypothesis while table 3 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 in the cases of *Method 1* and *Method 2*, we respectively have (1 B_ASR, 2 B_MT) and (2 B_ASR, 1 B_MT).

$e_{hyp_{slt}}$	surgeons	in	los	angeles	it	is	said
$e_{hyp_{mt}}$	surgeons	in	los	angeles	**	have	said
edit op.	Exact	Exact	Exact	Exact	Insertion	Substitution	Exact

Table 2. Example of edit distance between SLT and MT.

Algorithm 2 *Method 2* - Subtraction between SLT and MT errors

```
$list\_labels\_result \leftarrow empty\_list$
for each sentence $e_k \in \hat{e}_{slt}$ do
    $list\_labels\_sent \leftarrow empty\_list$
    for $j \leftarrow 1$ to $NumberOfWords(e_k)$ do
        if $label(e_{k_j})$ = 'G' then
            add 'G' to $list\_labels\_sent$
        else if $NameOfWordAlignment(l_{k_i})$ 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
```

f_{ref}	les chirurgiens	de	los	angeles	ont		dit
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 3. Example of quintuplet with 2-label and 3-label.

These differences are due to slightly different algorithms for label extraction. As Table 3 presents, "is" (SLT hypothesis) is aligned to "have" (MT hypothesis) and "have" (MT hypothesis) is labeled by "B". It can therefore be assumed 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 by B_ASR because the type of word alignment between "is" (SLT hypothesis) and "have" (MT hypothesis) is substitution (S), as shown in Table 2.

3.4 Statistics with 3-label Setting on the Whole Corpus

Table 4 presents 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.

As can be seen from Table 4, it is interesting to note that while 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 *ASR1* and *ASR2*. These statistics show that intersection between both methods is probably a good estimation of disentangled ASR and MT errors in SLT.

Task - *ASR1*	*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
Task - *ASR2*	***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

Table 4. Statistics with 3-label setting for *ASR1* and *ASR2*.

3.5 Qualitative Analysis of SLT Errors

Our new 3-label setting procedure allows us to analyze the behavior of our SLT system.

f_{ref}	peter frey est né le quatre août mille neuf cent cinquante sept à bingen
f_{hyp_1}	**pierre ferait aimé** le quatre août mille neuf cent cinquante sept à **big m**
f_{hyp_2}	**pierre** frey est né le quatre août mille neuf cent cinquante sept à **big m**
$e_{hyp_{mt}}$	peter frey was born on **4** august 1957 **to** bingen .
$e_{hyp_{slt1}}$	**pierre would liked the four** august **thousand nine hundred and fifty seven to big m**
$e_{hyp_{slt2}}$	**pierre** frey is born **the four** august **thousand nine hundred and fifty seven to big m**
e_{ref}	peter frey was born on august 4th 1957 in bingen .

Table 5. Example 1 - SLT hypothesis annotated with two methods - having a few *asr-errors*, a few *mt-errors* and many *slt-errors* such as 5 B_ASR1, 3 B_ASR2, 2 B_MT, 14 B_SLT1, 12 B_SLT2.

We can observe sentences with Table 5 presents, as an example, 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).

As shown in Table 6, on the contrary, there are many ASR errors leading to few SLT errors (ASR errors with few consequences such as morphological substitutions - for instance in French: de/des, déficit/déficits, budgétaire/budgétaires).

Finally, ASR errors as presented in Table 7 have different consequences on SLT quality (on a sample sentence, 2 ASR errors of system 1 and 2 lead to 14 and 9 SLT errors, respectively).

Figure 1 shows how our speech utterances are distributed in the two-dimensional (B_{ASR}, B_{MT}) error space.

4 Automatic Error Detection for SLT

In this paper, we use Conditional Random Fields (Lafferty et al., 2001) (CRFs) as our machine learning method, with WAPITI toolkit (Lavergne et al., 2010), to train our error detector based on MT and ASR engineered features. For ASR, we extract 9 features, which come from the ASR graph, from language model scores and from a morphosyntactic analysis. These detai-

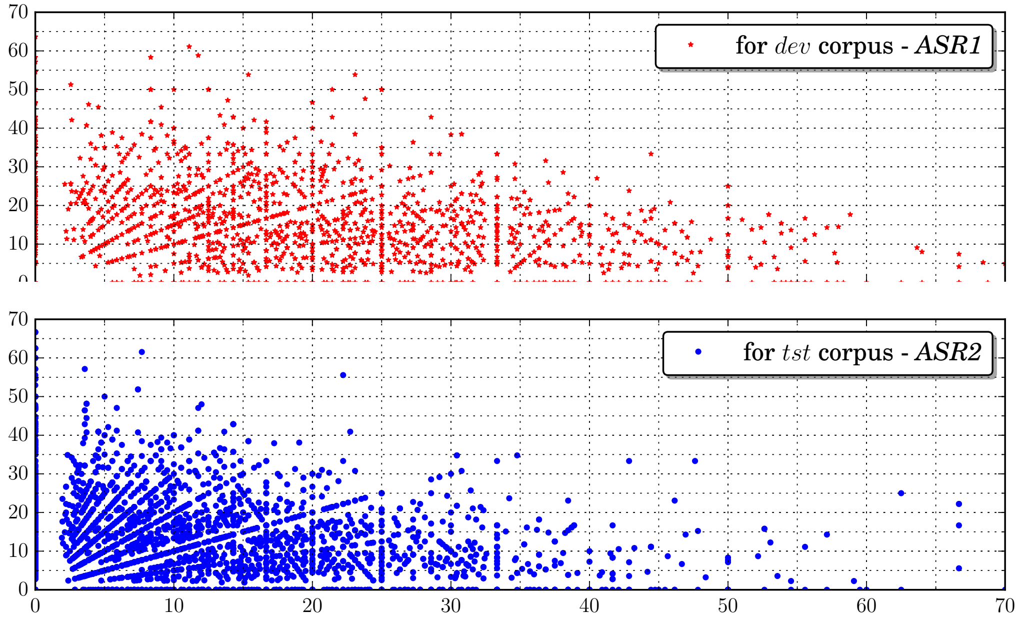


Figure 1. Example of the rate (%) of ASR errors (x-axis) versus (%) MT errors (y-axis) - for *dev/ASR1* and *tst/ASR2*.

f_{ref}	malheureusement le système européen de financement gouvernemental direct est
f_{hyp_1}	malheureusement le système européen financement **gouvernementale directe et**
f_{hyp_2}	malheureusement le système européen de financement gouvernemental direct est
$e_{hyp_{mt}}$	unfortunately , the european system of direct government funding is
$e_{hyp_{slt1}}$	unfortunately the european system direct government funding
$e_{hyp_{slt2}}$	unfortunately the european system of direct government funding is
e_{ref}	unfortunately , the european system of direct government funding is
f_{ref}	victime de la croissance économique européenne lente et des déficits budgétaires
f_{hyp_1}	**victimes** de la croissance économique européenne **venant de déficit budgétaire**
f_{hyp_2}	victime de la croissance économique européenne **venant** des déficits budgétaires
$e_{hyp_{mt}}$	a victim of european economic growth **slow** and budget deficits .
$e_{hyp_{slt1}}$	**and** victims of european economic growth **from** budget deficit
$e_{hyp_{slt2}}$	a victim of european economic growth **from the** budget deficits
e_{ref}	a victim of slow european economic growth and budget deficits .

Table 6. Example 2 - SLT hypothesis annotated with two methods - having many *asr-errors*, a few *mt-errors* and a few *slt-errors* such as 8 B_ASR1, 1 B_ASR2, 1 B_MT, 2 B_SLT1, 2 B_SLT2.

f_{ref}	nous ne comprenons pas ce qui se passe chez les jeunes pour qu' ils trouvent
f_{hyp_1}	nous ne comprenons pas **ceux** qui se passe chez les jeunes pour qu' ils trouvent
f_{hyp_2}	nous ne comprenons pas ce qui se passe chez les jeunes pour qu' **il trouve**
$e_{hyp_{mt}}$	we do not understand what is happening **among** young people for **that**
$e_{hyp_{slt1}}$	we do not understand **those who happens among** young people for **that**
$e_{hyp_{slt2}}$	we do not understand what is happening **among** young people
e_{ref}	we do not understand what is happening in young people 's mind for them
f_{ref}	amusant de maltraiter gratuitement un animal sans défense qui nous donne
f_{hyp_1}	amusant de **maltraité** gratuitement un animal sans défense qui nous
f_{hyp_2}	amusant de maltraiter gratuitement un animal sans défense qui nous donne
$e_{hyp_{mt}}$	**they are fun** to mistreat **free a** defenceless animal
$e_{hyp_{slt1}}$	**they** find **fun free** mistreated **a** defenceless animal
$e_{hyp_{slt2}}$	to find **it** amusing to mistreat **free a** defenceless animal
e_{ref}	to find amusing to mistreat defenceless animals without reason ,
f_{ref}	de l' affection de l' amitié et nous tient compagnie
f_{hyp_1}	de l' affection de l' amitié nous tient compagnie
f_{hyp_2}	de l' affection de l' amitié nous tient compagnie
$e_{hyp_{mt}}$	which gives us **the affection** , friendship and **keeps us airline** .
$e_{hyp_{slt1}}$	which **we affection of** friendship **we takes** company
$e_{hyp_{slt2}}$	which gives us **the affection of** friendship **we takes** company
e_{ref}	which gives us love , friendship and companionship .

Table 7. Example 3 - SLT hypothesis annotated with two methods - having the same number of *asr-errors*, but the different number of *slt-errors* extracted from *ASR1* and *ASR2* such as 2 B_ASR1, 2 B_ASR2, 12 B_MT, 14 B_SLT1, 9 B_SLT2.

led features could be found in (Besacier et al., 2014). For MT, we use a total of 24 major feature types which can be extracted with our word confidence estimation toolkit for MT (more details are given in (Servan et al., 2015)).

4.1 Experiments on 2-class Error Detection

Exp	MT+ASR feat. $p_{ASR}(q \mid x_f, f)^{\alpha}$ $*p_{MT}(q \mid e, f)^{1-\alpha}$	Joint feat. $p(q \mid x_f, f, e)$
F-avg1 (ASR1)	58.07%	64.90%
F-avg2 (ASR2)	53.66%	64.17%

Table 8. Error Detection Performance (2-label) on SLT ouptut for *tst* set (training is made on *dev* set).

In this experiment, 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-avg1* is obtained for SLT based on $ASR1$ whereas *F-avg2* is obtained for SLT based on $ASR2$. The classifier is evaluated on the *tst* part of our corpus and trained on the *dev* part.

We report in Table 8 the baseline error detection results obtained using both MT and ASR features for a 2-class problem (error detection). More precisely we evaluate two different approaches (*combination* and *joint*):

- First system (MT+ASR feat.) combines the output of two separate classifiers based on ASR and MT features. In this approach, 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 Equation 2 (we did not tune the α coefficient and set it *a priori* to 0.5).
- Second system (joint feat.) trains a single error detection system for SLT (evaluating $p(q|x_f, f, e)$ as in Equation 1 using joint ASR and MT features. ASR features are projected to the target words using automatic word alignments.

Table 8 shows that joint ASR and MT features improve error detection performance over the use of simple combination (MT+ASR). Based on this result, only the joint approach is used in our 3-class experiments of next section. We also observe that F-measure decreases when ASR WER is lower (*F-avg2<F-avg1* while $WER_{ASR2} < WER_{ASR1}$). So error detection for SLT might be more complicated as ASR system improves.

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

While the previous tables provided WCE performance for a single point of interest ($good/bad$ decision threshold set to 0.5), the curves of Figure 2 show the full picture of our WCE systems (for SLT) using speech transcriptions 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 threshold above 0.75, the use of both ASR and MT features is slightly beneficial. This is interesting because higher thresholds improves the F-measure on bad labels (so improves 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 threshold[4]).

4.2 Experiments on 3-class Error Detection

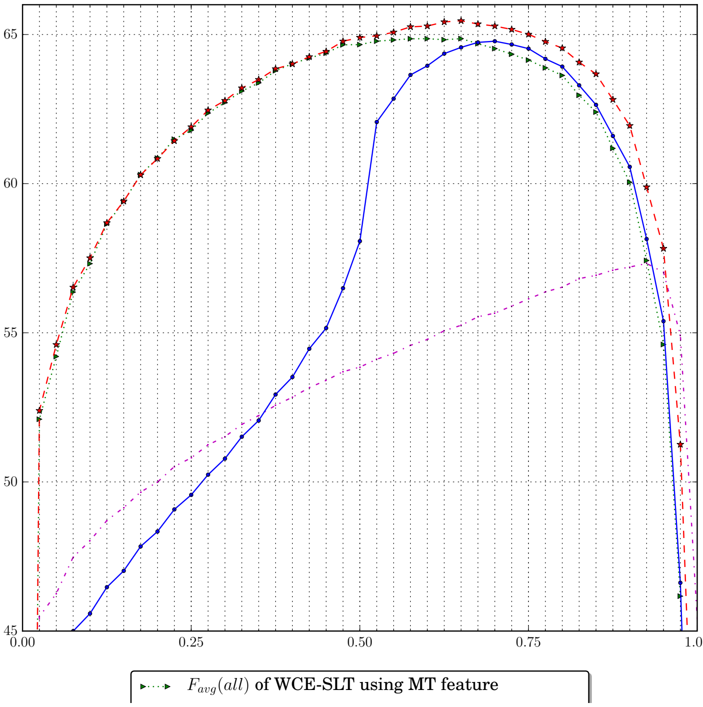
We report in Table 9 our first attempt to build an error detection system in SLT as a 3-class problem (*joint* approach only). We made our experiment by training and evaluating the model on *Intersection(m1, m2)* which corresponds to high confidence in the labels[5]. We compared two different approaches: *One-Step* is a single classifier for the 3-class problem while *Two-Step* first applies the 2 class (*G/B*) system and a second classifier distinguishes B_{ASR} and B_{MT} errors. Not much difference in F-measure is observed between both approaches. Table 10 also presents the confusion matrix between B_{ASR} and B_{MT} for the correctly detected (true) errors. Despite the relatively low F-scores of table 9, we see that our 3-labels classifier obtains encouraging confusion matrices in order to automatically disentangle B_{ASR} and B_{MT} on true errors.

5 Conclusions

This paper proposed to disentangle ASR and MT errors in speech translation. The binary error detection problem was recast as a 3-class labeling problem (*good, asr-error, mt-error*). First, two methods were proposed for the non trivial label setting and it was shown that both give

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

[5] However, we observed (results not reported here) 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.



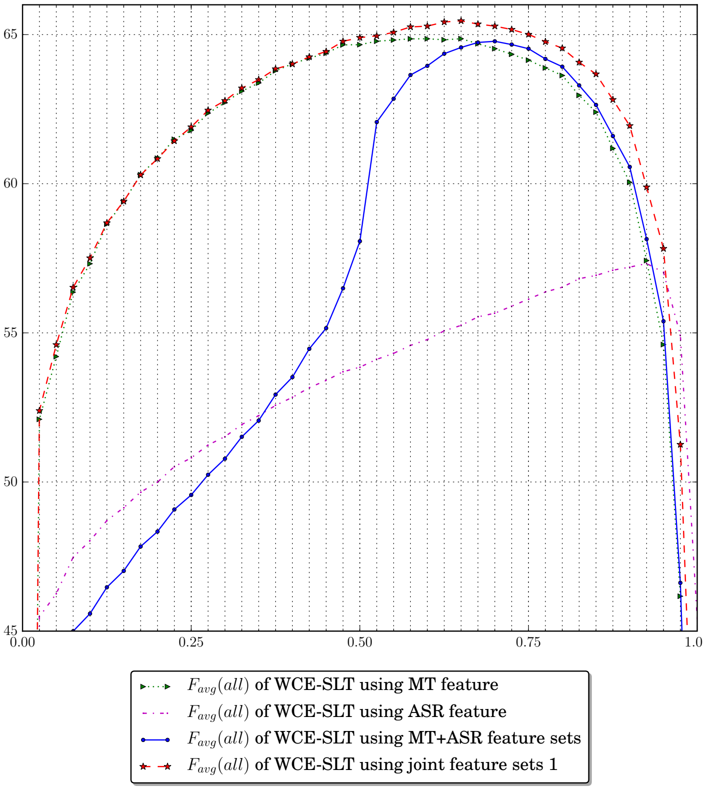


Figure 2. Evolution of system performance (y-axis - *F-mes1* - ASR1 and *F-mes2* - ASR2) for *tst* corpus (4050 utt) along decision threshold variation (x-axis) - training is made on *dev* corpus (2643 utt).

	2-class Full Corpus			**3-class Intersection Corpus (m1, m2)**			
				One-Step		**Two-Step**	
	ASR1	***ASR2***		***ASR1***	***ASR2***	***ASR1***	***ASR2***
F_G	81.79	83.17	F_G	85.00	85.00	84.00	85.00
F_B	48.00	45.17	F_{B_ASR}	44.00	42.00	44.00	42.00
			F_{B_MT}	14.00	15.00	16.00	17.00
F_{avg}	64.90	64.17	F_{avg}	47.67	47.33	48.00	48.00

Table 9. Error Detection Performance (2-label vs 3-label) on SLT output for tst set (training is made on *dev* set).

(1) Ref \ Hyp	***ASR1***		***ASR2***	
	B_ASR	***B_MT***	**B_ASR**	**B_MT**
B_ASR	*85.75%*	*14.25%*	81.57%	18.43%
B_MT	*44.46%*	*55.54%*	34.53%	65.47%
(2) Ref \ Hyp	***ASR1***		***ASR2***	
	B_ASR	***B_MT***	**B_ASR**	**B_MT**
B_ASR	*83.14%*	*16.86%*	80.02%	19.98%
B_MT	*49.41%*	*50.59%*	41.49%	58.51%

Table 10. Confusion Matrix on Correctly Detected Errors Subset for 3-class (1) One-Step; (2) Two-Step.

consistent results. Then, automatic detection of error types, using joint ASR and MT features, was evaluated and encouraging results were displayed on a French-English speech translation task. We believe that such a new task (not only detecting errors but also their cause) is interesting to build better informed speech translation systems, especially in interactive speech translation use cases.

Références

Besacier, L., Lecouteux, B., Luong, N. Q., Hour, K., and Hadjsalah, M. (2014). Word confidence estimation for speech translation. In *Proceedings of The International Workshop on Spoken Language Translation (IWSLT)*, Lake Tahoe, USA.

Besacier, L., Lecouteux, B., Luong, N.-Q., and Le, N.-T. (2015). Spoken language translation graphs re-decoding using automatic quality assessment. In *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Scotsdale, Arizona, United States.

Federico, M., Cettolo, M., Bentivogli, L., Paul, M., and Stüker, S. (2012). Overview of the IWSLT 2012 evaluation campaign. In *In proceedings of the 9th International Workshop on Spoken Language Translation (IWSLT).*

Galliano, S., Geoffrois, E., Gravier, G., Bonastre, J.-F., Mostefa, D., and 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)*, pages 315–320.

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 *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 177–180, Prague, Czech Republic.

Lafferty, J., McCallum, A., and Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting et labeling sequence data. In *Proceedings of ICML-01*, pages 282–289.

Lavergne, T., Cappé, O., and Yvon, F. (2010). Practical very large scale crfs. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 504–513.

Potet, M., Besacier, L., and Blanchon, H. (2010). The lig machine translation system for wmt 2010. In Workshop, A., editor, *Proceedings of the joint fifth Workshop on Statistical Machine Translation and Metrics MATR (WMT2010)*, Uppsala, Sweden.

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., and Vesely, K. (2011). The kaldi speech recognition toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society. IEEE Catalog No.: CFP11SRW-USB.

Servan, C., Le, N.-T., Luong, N. Q., Lecouteux, B., and Besacier, L. (2015). An Open Source Toolkit for Word-level Confidence Estimation in Machine Translation. In *The 12th International Workshop on Spoken Language Translation (IWSLT'15)*, Da Nang, Vietnam.

Snover, M., Madnani, N., Dorr, B., and Schwartz, R. (2008). Terp system description. In *MetricsMATR workshop at AMTA*.