An Open Source Toolkit for Word-level Confidence Estimation in Machine Translation

Christophe Servan†, Ngoc Tien Le†, Ngoc Quang Luong‡, Benjamin Lecouteux† and Laurent Besacier†

†GETALP – LIG, University of Grenoble Alpes, France
firstname.lastname@imag.fr
‡Idiap Research Institute, 1920 Martigny, Switzerland
nluong@idiap.ch

Abstract
Recently, a growing need of Confidence Estimation (CE) for Statistical Machine Translation (SMT) systems in Computer Aided Translation (CAT), was observed. However, most of the CE toolkits are optimized for a single target language (mainly English) and, as far as we know, none of them are dedicated to this specific task and freely available.

This paper presents an open-source toolkit for predicting the quality of words of a SMT output, whose novel contributions are (i) support for various target languages, (ii) handle a number of features of different types (system-based, lexical, syntactic and semantic). In addition, the toolkit also integrates a wide variety of Natural Language Processing or Machine Learning tools to pre-process data, extract features and estimate confidence at word-level. Features for Word-level Confidence Estimation (WCE) can be easily added / removed using a configuration file.

We validate the toolkit by experimenting in the WCE evaluation framework of WMT shared task with two language pairs: French-English and English-Spanish. The toolkit is made available to the research community with ready-made scripts to launch full experiments on these language pairs, while achieving state-of-the-art and reproducible performances.

1. Introduction
Statistical Machine Translation (SMT) has proven its efficiency during the last decade. For Computer Aided Translation (CAT) of documents, the following process is now broadly used: the SMT system produces raw translations then trained professional translators post-edit (correct) translation errors (PE). We believe that this SMT+PE pipeline can be improved using automatic confidence estimation (CE) where the system gives some clues about the quality of the SMT output. For instance, post-editors require to have information about the possible quality of the translation (Should they just post-edit the translation or rewrite the whole output? What are the main words/phrases they need to focus on?).

Building a method that could point out both correct and incorrect parts in SMT output is a key component to solve the above problems. When we limit the concept “parts” to “words”, the automatic confidence estimation process is called Word-level Confidence Estimation (WCE).

Past years have seen the emergence of shared tasks to estimate the translation quality (like WMT CE shared task†). In 2015, the organizers called for methods to predict the quality of SMT output at run-time on three levels: sentence-level (Task 1), word-level (Task 2) and (new) document-level (Task 3). This paper more precisely deals with the second task (WCE) but we believe it might be of interest to researchers who work in quality assessment for SMT.

Contributions Our experience in participating in task 2 (WCE) leads us to the following observation: while feature processing is very important to achieve good performance, it requires to call a set of heterogeneous Natural Language Processing tools (for lexical, syntactic, semantic analyses). Thus, we propose to unify the feature processing, together with the call of machine learning algorithms, to facilitate the design of confidence estimation systems. The open-source toolkit proposed (written in Python and made available on GitHub) integrates some standard as well as in-house features that have proven useful for WCE (based on our experience in WMT 2013 and 2014).

Outline The paper is organized as follow: Section 2 presents WCE task and related works on this topic. Section 3 is an overview of the features we extract while Section 4 describes the toolkit itself. Performances obtained using our WCE toolkit are given in Section 5 while Section 6 illustrates how one can easily apply feature selection for WCE using the provided code. Finally, Section 7 concludes this work and gives some perspectives.

2. WCE formalisation and related work
2.1. WCE formalisation
Machine translation (MT) consists in finding the most probable target language sequence \( \hat{e} = (e_1, e_2, ..., e_N) \) given a source language sentence \( f = (f_1, f_2, ..., f_M) \). We can represent Word-level Confidence Estimation (WCE) information as a sequence \( q \) (same length \( N \) of \( \hat{e} \)) where \( q = \)

†Since 2012 (http://www.statmt.org/wmt12/quality-estimation-task.html)
(q_1, q_2, ..., q_N) and q_i \in \{good, bad\}^2. Basically, the WCE component solves the equation:

\[
\hat{q} = \arg\max_q\{p(q|f, e)\} 
\]

This is a sequence labelling task that can be solved with several Machine Learning techniques such as Conditional Random Fields (CRF) [1]. However, to train sequence labelling models, we need a large amount of training data for which a triplet \((f, e, q)\) is available. In our case, we use binary labels associated to each word: Good or Bad to indicate whether a word is “correct” or “incorrect”, respectively.

2.2. Related work

According to [2], features for Word-level Confidence Estimation (WCE) can be classified in two types regarding their origin: the “external features” and the “internal features”. On the one hand, internal features are extracted from the SMT system itself like alignment table, N-best list, word graph, etc. On the other hand, external features mainly come from linguistic knowledge sources like syntactic parser, WordNet or BabelNet API, etc. In our approach, we use both types of features. They are mostly detailed in Section 3.

The first works about confidence estimation [3, 4], focused at the word level, was inspired by work done in automatic speech recognition [5]. The combination of a large amount of features, through a Naive Bayes model and a Neural Network, showed that Word Posterior Probability (WPP) was the most relevant internal feature. Later on, [6] integrated POS tagging and other external features. In the same way, [7] proposed 70 linguistic features for quality estimation at sentence level. Some of these features can be applied at word level. Their work also revealed the need of efficient machine learning algorithms to integrate multiple features and achieve better performance.

Recent workshops proposed some shared evaluation tasks of WCE systems, in which several attempts of participants to mix internal and external features were successful. The estimation of the confidence score uses mainly classifiers like Conditional Random Fields [8, 9], Support Vector Machines [10] or Perceptron [11].

Further, some investigations were conducted to determine which feature seems to be the most relevant. [10] proposed to filter features using a forward-backward algorithm to discard linearly correlated features. Using Boosting as learning algorithm, [2] was able to take advantage of the most significant features.

Our work, inspired by all those previous papers, proposes to mix internal and external features and uses CRF as decision algorithm to estimate a WCE score. The technical novelty is their integration in a single toolkit, with ready-made scripts, to quickly run reproducible experiments on different language pairs. It also provides a built-in feature selection approach. Contrarily to the toolkit proposed in [12], our framework allows a quick and easy reproduction of the results presented in this paper and addition of new features is straightforward.

3. Available Features

Our toolkit extracts several internal and external features to train a classifier, as indicated in Table 1. These features were chosen because of their relevance in previous Word-level Confidence Estimation tasks [13, 14, 15]. Some of them are already described in detail in some previous papers [5, 3, 4, 6, 10, 2, 16]. Consequently, the novel features, which we added into our current toolkit, are in “bold” in Table 1. Also, the features in “italic” are conventional features but extracted using a new approach.

The feature list could be extended (by us or by other contributors) in the future, since the toolkit is made available to the research community. For instance, we plan to integrate the use of monolingual or bilingual word embeddings following the works of [17].

It is important to note that our toolkit extracts the features regarding tokens in the machine translation (MT) hypothesis sentence. In other words, one feature is extracted for each token in the MT output. So, in the Table 1, target refers to the feature coming from the MT 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.

3.1. Internal Features

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

In addition to features corresponding to source / target words or POS (feat. 5 to 10), Word Posterior Probability (WPP), WPP Max, WPP Min and Nodes features are extracted from a confusion network, which comes from the output of the machine translation N-best list. WPP Exact is the WPP value for each word concerned at the exact same position in the graph. WPP Any extract 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.

In the example shown in Figure 1, the target word “function” gets a WPP Exact at 0.2, WPP Min at 0.1 and WPP Max at 0.4.
Table 1: Features extracted by the toolkit: highlights in “bold” are the new features we propose, the other features are those classically extracted ; we put in “italic” those for which we propose a new extraction method compared to previous work (see Section 4.2.3). Features indicated with “∗” are internal ones.

### 3.2. External Features

Below is the list of the external features we use in our toolkit:

- **Proper Name**: indicates if a word is a proper name (same binary features are extracted to know if a token is Numerical, Punctuation or Stop Word).
- **Unknown Stem**: informs whether the stem of the considered word is known or not.
- **Number of Word/Stem Occurrences**: count the occurrences of a word/stem in the sentence.
- **Alignment context features**: these features (#11-13 in Table 1) are based on collocations and proposed by [18]. Collocations could be an indicator for judging if a target word is generated by a particular source word. We also apply the reverse, the collocations regarding the source side:
  - **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.

With the example presented in Table 2, the target word “of” is aligned with “de”. The source context extracted corresponds to the two words around “de”, which are “nature” and “l’”. The source alignment context features are “of/nature”, “of/de” and “of/l’”. In the same way, the target alignment context features of “de” are: “de/nature”, “de/of” and “de/the”.

We applied the same context extraction for Part-of-Speech and Stems.

Table 2: Example of parallel sentence where words are aligned one-to-one.

<table>
<thead>
<tr>
<th>Target</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>the</td>
<td>is</td>
</tr>
<tr>
<td>nature</td>
<td>nature</td>
</tr>
<tr>
<td>of</td>
<td>de</td>
</tr>
<tr>
<td>the</td>
<td>l’</td>
</tr>
<tr>
<td>independence</td>
<td>independence</td>
</tr>
<tr>
<td>granted</td>
<td>octroye</td>
</tr>
</tbody>
</table>

- **Longest Target (or Source) N-gram Length**: we seek to get the length \( (n + 1) \) of the longest left sequence \( (w_{i-n}, \ldots) \) 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 \( w_i-n, w_i-n+1, \ldots \) 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.

- The word’s constituent label (Constituent Label) and its depth in the constituent tree (Distance to Root) are extracted using a syntactic parser, the Figure 2 illustrates the distance between a word and its root in the tree. In the case of “working”, the Constituent Label is VBG and the Distance to Root value is 6.

- **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 (optionnally) test the presence of the target word in online translations given respectively by Google Translate and Bing Translator.
Figure 3: Pipeline of our Word-level Confidence Estimation tool

4. Toolkit

In this section, we detail our toolkit, which is a complete out-of-the-box Word-level Confidence Estimation (WCE) system. It is a customizable, flexible, and portable platform.

4.1. Pipeline Overview

Our toolkit is described in Figure 3. It contains three essential components: preprocessing, feature extraction and training / labeling. It integrates several existing Natural Language Processing (NLP) tools and API. It is developed in Python 3 to use efficiently existing libraries/toolkits as well as being object-oriented designed.

The source code is available on a GitHub repository and provided with ready-made scripts to run reproducible experiments on a French–English WCE task (for which the data is also made available).

4.2. System Design

The first steps are the preprocessing and the feature extraction during which the toolkit processes and adds information to the initial corpora available. Then, the most important step consists of training a classifier using the features extracted (training phase) or in the labelling of the test corpus (decoding phase).

We also added a threshold optimization and a feature selection phase which are later described (see Sections 5.5 and 6 respectively for threshold optimization and feature selection).

All these phases can be parameterized using a single configuration file.

4.2.1. Configuration file

A configuration file gathers the main WCE parameters. It is stored in YAML format. The main configuration parameters concern the source and target languages involved and the path to the input corpus and its translation.

4.2.2. Preprocessing Phase

Preprocessing consists of obtaining POS tags, word alignments and all needed analyses from the available parallel corpus (the target being a MT output made up of raw text – 1-best and N-best of MT). First, input data is lowercased and/or tokenized if necessary. Then, TreeTagger toolkit [19] is applied to get the Part-Of-Speech (POS) tags and stem of each word in both source and target languages. The different POS extracted are normalized. Finally, word alignments are obtained using GIZA++ [20].

4.2.3. Features Extraction

As said before, the internal features come from the output of the Statistical Machine Translation (SMT) system. In this part we mainly focus on the extraction of the external features, given by toolkits which are not part of the SMT system.

The TreeTagger toolkit [19] is involved in the extraction of the following features: “Proper Names”, “Unknown Stems” and “Source/Target Stem”. GIZA++ [20] helps us to extract the context alignment features for POS, Word and Stems. To compute the features “Longest Target N-gram Length” and “Longest Source N-gram Length” we use the SRILM toolkit [21]. The word’s constituent label (“Constituent Label”) and its depth in the constituent tree (“Distance to Root”) are also extracted using Bonsai (for French) [22, 23] or Berkeley parser (for other languages) [24]. To represent hierarchical structures and extract the two features, the Natural Language ToolKit (NLTK) [25] in Python is used. The BabelNet [26] API is used to extract the feature “Target polysemy count”.

Finally, the features “Occurences in Google Translate” and “Occurences in Bing Translator” are extracted by using the Google Translate and Bing Translator API, respectively.

4.2.4. Training / Decoding Phase

Once the final feature extraction stage has been completed, we use Conditionnal Random Fields (CRF) as machine learning technique through the Wapiti toolkit [27].

The classifier uses all the chosen features and it is trained on a preliminary labelled corpus (see next section for example of corpora directly usable with our toolkit). During decoding phase, the classifier determines, from a test corpus, whether a word should be labelled as “correct” or “incorrect” (respectively Good or Bad).
5. WCE Experiments

This section presents the experiments done for 2 different language pairs: French–English (fr–en) with the corpus provided by [28] and English–Spanish (en–sp) corresponding to the WMT shared task on word confidence estimation (2014 edition6).

5.1. The French–English post-edited corpus

The fr–en corpus contains 10881 translations. It was taken from several French–English news corpora from former WMT evaluation campaigns (from 2006 to 2010) [28].

To obtain the translations, [28] used a French–English phrase-based translation system based on the Moses toolkit [29]. This medium-sized system was trained on Europarl and News parallel corpora for a former WMT evaluation shared-task (system more precisely described in [30] – 1.6M parallel sentences and 48M monolingual sentences in target language).

The hypotheses translated were post-edited according to the methodology described in [28]. 10000 random sentences were extracted to create the training data and the remaining sentences were used for the evaluation corpus.

In order to evaluate our Word-level Confidence Estimation (WCE) system, we obtained a sequence $q$ of quality labels (recall that $q = (q_1, q_2, ..., q_N)$ and $q_i \in \{\text{good, bad}\}$) using TER-Plus toolkit [31]. Each word or phrase in the hypothesis $e_{hyp}$ is aligned to a word or phrase in the reference ($e_{ref}$) with different types of edit: “I” (insertions), “S” (substitutions), “T” (stem matches), “Y” (synonym matches), “P” (phrasal substitutions) and “E” (exact match). Then, we re-categorize the obtained 6-label set into binary set: the “E”, “T” and “Y” belong to the good (“G”), whereas the “S”, “P” and “I” belong to the bad (“B”) category.

An example of output of TER-Plus evaluation tool is shown in Table 3.

5.2. Adaptation to a new language pair

To evaluate our toolkit on another language pair (English–Spanish), we used the official data from WMT 2014 shared task on WCE.

One of the strength of our toolkit is the easiness to adapt it to another language pair within the (so-far) supported languages which are French, English, and Spanish. Thus, a few configuration parameters were changed to move from the French–English (fr–en) to English–Spanish (en–es), which are mainly the source language, the target language, and paths associated to input files.

Consequently, our WCE toolkit process en–es task in the same way as for fr–en task, but some features may not be extracted due to language-pair specificities: unavailable tools, no N-best, etc. For instance, for the en–es task, since the N-best list is not available, we cannot extract the five following internal features: “WPP Exact”, “WPP Any”, “Nodes”, “WPP Min” and “WPP Max”.

5.3. Results

The WCE evaluation measures are the Precision ($P$), the Recall ($R$) and the F-Measure ($F$) of each label (as reminder, the decision label can be either good or bad). We use wapiti [27] to train the CRF model and label the words.

5.4. Comparison with the State-of-the-Art

Table 4 presents the results obtained in the WMT14 Quality Estimation shared task with the language pair English–Spanish (en–es).

Table 4: Results of the best systems at the Word-level Quality Estimation task (en–es) at WMT14 [15], only the Mean F-Measure ($M$-$F$) and the F-Measure ($F$) on the bad labels are available to compare the performances of our toolkit.

Using the default decision threshold of our classifier, the Table 4 presents the results obtained in the WMT14 Quality Estimation shared task with the language pair English–Spanish (en–es).

The results show that our toolkit obtained similar performances compared to the State-of-the-Art. We could not compare with the CE toolkit mentioned in [12] since they did not provide full results within the framework of the WMT14 evaluation. Future work could involve a comparison between our toolkit and the toolkit presented in [12].

5.5. Decision threshold optimization

Table 5 shows the classification performances of our toolkit for the two different language pairs: the French–English (fr–en) and the English–Spanish (en–es). The latter corresponds to the Quality Estimation shared task of WMT14 [15].

Our toolkit proposes to optimize the decision threshold but, in this context, what we report can be only considered as an oracle threshold setting since no real development corpus was available for both language pairs. These results are only reported to demonstrate the ability of the toolkit to tune the decision threshold. With this optimization, the scores are improved for the bad label (+2.89 points) regarding the results obtained with the default threshold in the fr–en task. In the en–es task, the oracle threshold slightly improves the results, according to the Mean F-Measure (+0.11 points).

---

6 http://www.statmt.org/wmt14/quality-estimation-task.html
The following feature sets were evaluated in this section:

6.1. Experimenting with different feature sets

Task with the default decision threshold.

The next experiments reported were done for the fr–en selection and analysis of performance with different feature sets. This section illustrates how the toolkit can be used for feature selection and analysis of performance with an oracle threshold (to optimize Mean F-measure of Good and Bad labels).

### 6. Features selection

This section illustrates how the toolkit can be used for feature selection and analysis of performance with different feature sets. The next experiments reported were done for the fr–en task with the default decision threshold.

#### 6.1. Experimenting with different feature sets

The following feature sets were evaluated in this section:

- the baseline features (Base.) given in Table 1 (not “bold”, not “italic”, no feat. 28-29),
- same as above + modified features estimated with a new method (in “italic” in Table 1) are added (mod.) ;
- same as above + the new features (new) mentioned in Table 1 (the ones in “bold”) ;
- same as above + features 28-29 of Table 1 involving online MT systems (MT).

#### 6.2. Feature selection using Sequential Forward Selection (SFS) algorithm

Going further, we propose to process a finer feature selection using the Sequential Forward Selection (SFS) algorithm for which scripts are made available in our toolkit distribution.

While feature selection can be made through several approaches [34], we chose to use the SFS method. It is a bottom up algorithm which starts from a feature set noted $Y_k$ (which can be empty or not) and selects as first feature $(x)$ the one that maximizes the Mean F-Measure, $M F(Y_k + x)$, from a set of features $(J_k)$. The algorithm below summarizes the whole process:

```
while size of $J_k > 0$ do
    maxval = 0
    for $x \in J_k$ do
        if maxval < $M F(Y_k + x)$ then
            maxval ← $M F(Y_k + x)$
            bestfeat ← $x$
        end if
    end for
    add bestfeat to $Y_k$
    remove bestfeat from $J_k$
end while
```

In Table 7 we present the result of the SFS algorithm, which ranks our new features starting from an empty feature set. The dash line marks the limit of the best feature set according to the Mean F-Measure (with 65.14 points).

It appears that most of new features we added (in “bold”) bring relevant information associated to classical ones (no highlight and in “italic”). Only the feature “Target Stem” seems to be irrelevant for the fr–en task. One reason for that might be that for the English language, stem and words features may be highly correlated.

#### Table 5: The toolkit’s WCE performances with fr–en and en–es (WMT14) tasks. Note that for each language pair, the first block of results corresponds to the performance obtained with default decision threshold and the second block corresponds to the performance with an oracle threshold (to optimize Mean F-measure of Good and Bad labels).

<table>
<thead>
<tr>
<th>Task</th>
<th>Threshold</th>
<th>Label</th>
<th>P</th>
<th>R</th>
<th>F</th>
<th>M-F</th>
</tr>
</thead>
<tbody>
<tr>
<td>fr–en</td>
<td>Default</td>
<td>Good</td>
<td>84.45</td>
<td>90.22</td>
<td>87.24</td>
<td>64.96</td>
</tr>
<tr>
<td></td>
<td>Bad</td>
<td>50.10</td>
<td>37.16</td>
<td>42.67</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Optimized</td>
<td>Good</td>
<td>85.50</td>
<td>85.83</td>
<td>85.64</td>
<td>65.92</td>
</tr>
<tr>
<td></td>
<td>Bad</td>
<td>45.61</td>
<td>45.50</td>
<td>45.56</td>
<td></td>
<td></td>
</tr>
<tr>
<td>en–es</td>
<td>(WMT14)</td>
<td>Default</td>
<td>71.24</td>
<td>71.73</td>
<td>74.35</td>
<td>60.76</td>
</tr>
<tr>
<td></td>
<td>Bad</td>
<td>51.82</td>
<td>43.28</td>
<td>47.17</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Optimized</td>
<td>Good</td>
<td>71.42</td>
<td>73.82</td>
<td>74.03</td>
<td>60.87</td>
</tr>
<tr>
<td></td>
<td>Bad</td>
<td>51.49</td>
<td>44.45</td>
<td>47.71</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Table 6: Improvements obtained regarding the features added. For both labels (Good and Bad) we use the Precision (P), Recall (R) and F-Measure (F). The Mean F-Measure of Good and Bad labels is presented in the last column.

<table>
<thead>
<tr>
<th>Features</th>
<th>Label</th>
<th>P</th>
<th>R</th>
<th>F</th>
<th>M-F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base.</td>
<td>Good</td>
<td>81.97</td>
<td>92.22</td>
<td>86.80</td>
<td>58.64</td>
</tr>
<tr>
<td></td>
<td>Bad</td>
<td>44.17</td>
<td>23.28</td>
<td>30.48</td>
<td></td>
</tr>
<tr>
<td>+ mod.</td>
<td>Good</td>
<td>83.21</td>
<td>90.99</td>
<td>86.92</td>
<td>62.00</td>
</tr>
<tr>
<td></td>
<td>Bad</td>
<td>47.24</td>
<td>30.53</td>
<td>37.09</td>
<td></td>
</tr>
<tr>
<td>+ new</td>
<td>Good</td>
<td>83.55</td>
<td>90.11</td>
<td>86.70</td>
<td>62.65</td>
</tr>
<tr>
<td></td>
<td>Bad</td>
<td>46.75</td>
<td>32.86</td>
<td>38.60</td>
<td></td>
</tr>
<tr>
<td>+ MT</td>
<td>Good</td>
<td>84.45</td>
<td>90.22</td>
<td>87.24</td>
<td>64.96</td>
</tr>
<tr>
<td></td>
<td>Bad</td>
<td>50.10</td>
<td>37.16</td>
<td>42.67</td>
<td></td>
</tr>
</tbody>
</table>

#### Table 7: Rank of each feature according to the Sequential Forward Selection algorithm within the framework of the fr–en task. The Dash line marks the best Mean F-Measure score obtained with 65.14 points.

<table>
<thead>
<tr>
<th>Rank</th>
<th>Feature</th>
<th>Rank</th>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Stem context Alignments</td>
<td>16</td>
<td>Stop Words</td>
</tr>
<tr>
<td>2</td>
<td>WPP Exact</td>
<td>17</td>
<td>Nodes</td>
</tr>
<tr>
<td>3</td>
<td>Word context Alignments</td>
<td>18</td>
<td># of Stem Occurrences</td>
</tr>
<tr>
<td>4</td>
<td>WPP Max</td>
<td>19</td>
<td>Numeric</td>
</tr>
<tr>
<td>5</td>
<td>WPP Any</td>
<td>20</td>
<td>Unknown Stem</td>
</tr>
<tr>
<td>6</td>
<td>WPP Min</td>
<td>21</td>
<td>Target Word</td>
</tr>
<tr>
<td>7</td>
<td>POS context Alignments</td>
<td>22</td>
<td>Source POS</td>
</tr>
<tr>
<td>8</td>
<td>Occur in Google Translate</td>
<td>23</td>
<td>Polysemous Count – Target</td>
</tr>
<tr>
<td>9</td>
<td>Longest Target N-gram Length</td>
<td>24</td>
<td>Source Word</td>
</tr>
<tr>
<td>10</td>
<td>Occur in Bing Translate</td>
<td>25</td>
<td>Constituent Label</td>
</tr>
<tr>
<td>11</td>
<td>Source Stem</td>
<td>26</td>
<td>Punctuation</td>
</tr>
<tr>
<td>12</td>
<td>Target Backoff Behaviour</td>
<td>27</td>
<td># Target Stem</td>
</tr>
<tr>
<td>13</td>
<td>Longest Source N-gram Length</td>
<td>28</td>
<td>Proper Name</td>
</tr>
<tr>
<td>14</td>
<td># of Word Occurrences</td>
<td>29</td>
<td>Target POS</td>
</tr>
<tr>
<td>15</td>
<td>Distance To Root</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

We can observe for all the steps a general improvement of the Mean F-Measure in Table 6. The baseline is 58.64, while the use of modified features enables to get over 62. The new features show their usefulness with a Mean F-Measure score at 62.65 points. Finally, adding occurrences coming from on-line Machine Translation systems enables us to get 64.96 points. Even if using online MT systems for WCE can appear as controversial, this seems to bring useful information to our classifier.
7. Conclusion and Perspectives

This paper presented our Word Confidence Estimation (WCE) approach made available through an open-source toolkit. It combines some classical features as well as some new in-house features. All these features are passed through a Conditional Random Fields (CRF) classifier to estimate the correctness of a word.

The WCE experiments conducted achieve State-of-the-Art and reproducible performances measured on two different data sets corresponding to two language pairs (French–English and English–Spanish). Thanks to its flexibility, our toolkit is nearly language independent, as long as the user can provide grammars and models for the specified languages.

Our WCE toolkit has been packaged and released for others to be able to reproduce rapidly the experiments reported in this article. This package is made available on a GitHub repository under the licence GPL V3.

In addition to this toolkit, comes a special module, which enables feature selection automatically using SFS algorithm (sequential forward selection). A more performant algorithm will be added in the near future like the Sequential Floating Forward Selection algorithm, which has backtracking capabilities.

Further work will focus on (i) adding features (based on word embeddings for instance) and (ii) evaluating the toolkit efficiency in a real Computer Assisted Translation (CAT) framework. We also plan to extend our toolkit to the design of WCE for speech recognition and speech translation tasks.

8. Acknowledgement

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper.

This work was partially founded by the French National Research Agency (ANR) through the KEHATH Project.

9. References


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