# SMT Project Proposal: MWEs in Machine Translation

Jakub Waszczuk

January 2019

## 1 Background

While the quality of industrial-strength statistical/neural machine translations systems (MT) constantly improves, it's still far from perfect. Clearly one of the challenges is the capability of MT systems to automatically „learn" from raw training data and much of the current research is targeted at this issue. Another challenge is how to incorporate background knowledge in the form of dictionaries or context-specific rules, which can help in dealing with domain-specific terminology (industry jargon, product names, etc.). For instance, Systran recently added support for such user-specified dictionaries.[1]

From the linguistic perspective, the latter challenge is much related to the so-called *multiword expressions* (MWEs). MWEs are expressions whose meaning is non-compositional, i.e., cannot be easily derived from the meaning of their component words. Hence, their translation cannot be done on word-by-word basis either. MWEs include (multiword) technical terms, various classes of named entities – products, places, organizations, persons, etc. – as well as verbal expressions – idioms (*to* ***kick the bucket***), verb-particle construction (*to* ***blow up***), light-verb constructions (*to* ***pay*** *a* ***visit***), etc.[2] To make things worse, such expressions can easily be discontinuous or nested (***take*** *the fact that I didn't* ***give up into account***). All this makes them difficult to process and, in particular, to translate.

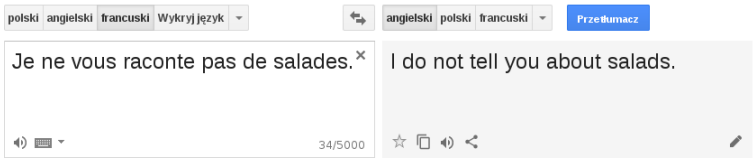


Figure 1: Example of an incorrect translation of the French idiomatic expression *raconter des salades*, 'to tell stories'

The behaviour of the current MT systems on MWEs is often sub-optimal. This may be surprising, since MWEs are relatively easy to identify due to their lexical nature, i.e., the fact that they contain very specific component words.

[1] `http://blog.systransoft.com/our-neural-network-just-learned-syntax/`
[2] See `https://parsemefr.lif.univ-mrs.fr/parseme-st-guidelines/1.1` for more.

Yet even the state-of-the-art neural MT tools struggle with translation of many MWEs (see Fig. 1 for an example).

## 2 Goal

**Design.** The goal of this project is to design an MT architecture which would be able to better handle MWEs. One possible way is to allow the MT user to specify custom dictionaries (just as proposed by Systran), but you can think of other approaches. Here are some of the questions you would have to consider to successfully finish the project:

- How can such a custom dictionary look like (**format**)?
- How can it be plugged in the process of translation? In particular, how to identify the instances of the dictionary entries present in a given input sentence (**identification**)?
- Statistical/neural MT systems model the probability $P(\boldsymbol{y}|\boldsymbol{x})$ of translating input sentence $\boldsymbol{x}$ to output sentence $\boldsymbol{y}$. How to incorporate the identified entries in the translation probability formula (**model**)?
- Does the translation algorithm have to be adapted? If so, would the modification change its computational complexity (**decoding**)?
- How to make sure that the additional dictionaries do not decrease the overall quality of the translation system (**regression testing**)?

You can assume the standard phrase-based translation model as a point of departure. However, if you are feeling curious/adventurous, you can also consider the questions given above in the context of neural machine translation.

**Proof of concept.** The second part of the project is to realize an MT prototype implementing (some of) the solutions proposed for the questions above. To this end, you can rely on the code we developed during the practical sessions. You don't have to implement all the ideas proposed in the **design** stage, focus on the simple ones so as to finish in time prescribed for the project.

## 3 Deliverables

You will be expected to provide the following deriverables at the end of the project (the deadlines will depend on when exactly you subscribe).

- **Report.** A clear description of the design of the MT system, including the proposed answers to the question from Sec. 2. The report should also indicate the workload – how each member of the group contributed to the project. There is no limit on the number of pages, but the suggested number for this kind of a project is between 5 and 25.
- **Implementation.** A clear and documented source code, with installation instructions and test data. A short description explaining which of the ideas from the **design** section were implemented should be also reported.

Once you have a clear picture concerning your design, we shall have a **half-way meeting** where you explain your main ideas and your implementation plans. There will also be a **final meeting** where you will have a chance to present your work. On request, we can also have other, short informal meetings to discuss your progress.