Project Work
Knowledge Engineering
Translation Coherence

Andrea Lavista
andrea.lavista@studio.unibo.it

November 25, 2021

Contents

Chapter 1

Introduction

1.1 Recap of the previous project

Let's do at first a short recap of those aspects of the previous project (realised with Lorenzo Amorosa and Michele Iannello) that have been used as basis of the activities performed in the project work.
As result of the previous project we produced a set of procedures that allows to catch differences generated by translations. In particular, these were the main steps:

- selecting English sentences from a corpus (in our case EuroParl dataset)
- generating with a machine translator different versions of those sentences, translating the initial sentences towards some languages (we chose German, Italian, Chinese) and then back translated to English
- generating with a machine reader (FRED) the knowledge graphs that represent the sentences
- performing a comparisons between graphs using preset rules to detect relations like equivalences, differences, synonyms, etc.
- generating knowledge graphs that contains the triples that describe the results of the comparison.

Now, I'll briefly explain the criteria for some of the predicates of the comparison graphs that I used in the project work.

Given two knowledge graphs generated by the two sentences their individuals are connected through these predicates:

- **equivalent**: if they have the same lemma and at least 1 neighbour "in common"
- **synonymy**: if they have synonymous lemmas and at least 1 neighbour "in common". Example: "...the **people** in a number of countries..." "...the **populations** of some countries...".
- **different**: if they have different (and not synonymous) lemmas and at least 2 neighbours "in common". Example: ...the session of the European Parliament **adjourned** on Friday..." "...the session of the European Parliament, which was **interrupted** on Friday...".
- **differentContext**: same lemma and 0 neighbours "in common"

Then the elements not classified are marked with the predicate **onlyIn**. For the classes we perform a similar analysis watching at their name, present in the URI.

1.2 Error analysis

Then I did a small analysis to understand which were the the most recurrent errors in the output. These are the main weaknesses:

- not all synonyms are captured
- errors when an expression in the initial version map into another one with different length. Examples:
  - **ensure =>make sure**: "ensure" and "make" are marked as synonyms while sure is classified as something present only in the second sentence
  - **in fact =>indeed**: "in" doesn't appear in the knowledge graph generated with Fred and "fact" and "indeed" are not classified
- different relations issue:
  - **put it to a vote =>vote on it**: "put" and "vote" are marked as different in the sentences "We then put it to a vote", "Then we vote on it".

– due to a different order of the constituent that led to a different construction of the graph "there" and "time" are connected with the different predicate in the sentences "**There** has therefore been enough time for the Commission...", "So the Commission had enough **time** to...".

1.3 Activities for the project work

Here's the list of activities planned and implemented for the project work:

- Automatization of the translation step (with DeepL and Argos Translate)
- Automatization of the generation of the knowledge graphs using the FRED's API
- Definition of metrics to assess the quality of the translations based on the predicates we defined in the project
- Statistical analysis on the metrics results based on different aspects like languages, sentence lengths, and others.

The first two tasks are pretty straightforward: I implemented them using Python language and this steps were necessary to apply the comparisons to a large amount of sentences. So, after implementing it, I generated the knowledge graphs representing sentences from EuroParl and WikiNER datasets.
In the next chapter I'll talk about the analysis performed explaining the metrics used when needed.

Chapter 2

Analysis by language

The idea of this analysis is to see how the choice of the pivoting language in the process of the back and forth translation affects the quality of the new versions generated (i.e. how many differences causes with respect to the original text).
The dataset used is composed in this way:

- 494 English sentences from the Europarl dataset
- 576 English sentences from WikiNER dataset

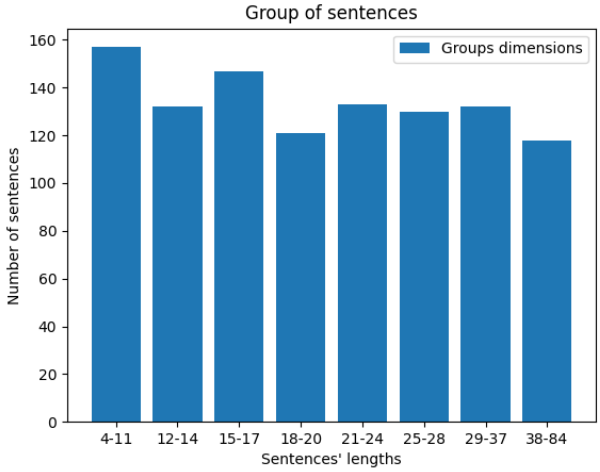

Figure 2.1: The distribution of the sentences respect to their lengths

These sentences were translated towards 6 languages and then back translated to English, generating 1170 sentences for each language. The chosen

languages are: German (DE), Dutch (NE), Italian (IT), Finnish (FI), Chinese (ZH), Korean (KO). German and Dutch are both Indo-european and germanic languages as well as English, Italian and Finnish are both Indo-european languages as well as English but from different branches than germanic, while Chinese and Korean are non Indo-european languages. Moreover, Finnish and Korean are agglutinative languages. The choice fo the languages was done in order to have languages closer and further to English from both a genealogical and a typological (respect to morphology) point of view.

2.1 BLEU Score

Before computing my metric on these data I computed the BLEU score (a well known metric for machine translation) on the sentences. Here's the results.
EuroParl + WikiNER BLEU score average:
IT = 0.525 NL = 0.373 DE = 0.338
FI = 0.204 KO = 0.188 ZH = 0.136

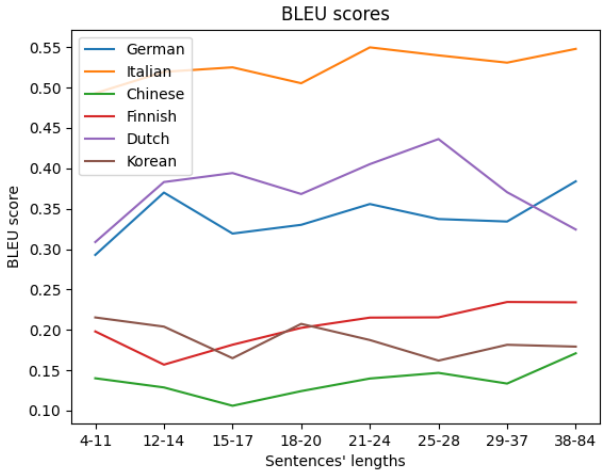

Figure 2.2: BLEU score applied to EuroParl + WikiNER dataset

Italian is by far the language with highest score, then Dutch and German which have similar scores and with lowest scores the others three.
Then I computed the BLEU score upon the 2 part of the dataset separately,

EuroParl before and then WikiNER, in order to see if there were noticeable differences between the two.
EuroParl BLEU score average:
IT = 0.490 DE = 0.375 NL = 0.308
FI = 0.184 ZH = 0.175 KO = 0.164

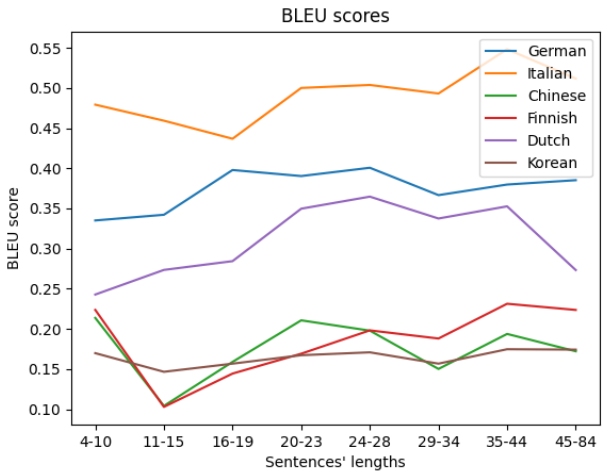

Figure 2.3: BLEU score applied to EuroParl dataset

WikiNER BLEU score average:
IT = 0.556 NL = 0.429 DE = 0.307
FI = 0.221 KO = 0.208 ZH = 0.102

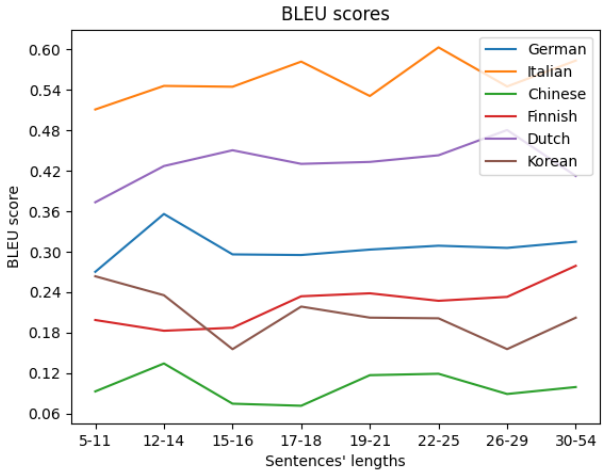

Figure 2.4: BLEU score applied to WikiNER dataset

We can see here that German is better than Dutch in EuroParl and worst than Dutch in WikiNER. Also Chinese has simialr score with Finnish and Korean in the case of EuroParl but clearly worst than those two in WikiNER. This is probably due to the characteristic of the dataset used to train machine translator: amount of data used and its representativeness, and its similarity with the sentences used in this project.

2.2 Equivalent/synonymy metric

Now I'll talk about a metric I defined based on the predicates of the comparison analysis performed on the knowledge graphs. Given that the comparison graphs are composed by predicates connecting elements of the first sentences to the ones of the second sentences, we can discriminate the predicates that indicate a good translation to the ones that indicate a bad translation. So I defined this metric as the number of equivalent/synonymy predicates (which denote a good translation) divided by the total number of predicates.

$$E/S\ score = \frac{number\ of\ equivalent/synonymy\ predicates}{total\ number\ of\ predicates} \tag{2.1}$$

EuroParl + WikiNER Equivalent/synonymy metric average:
IT = 0.689 NL = 0.560 DE = 0.506
FI = 0.380 KO = 0.421 ZH = 0.359

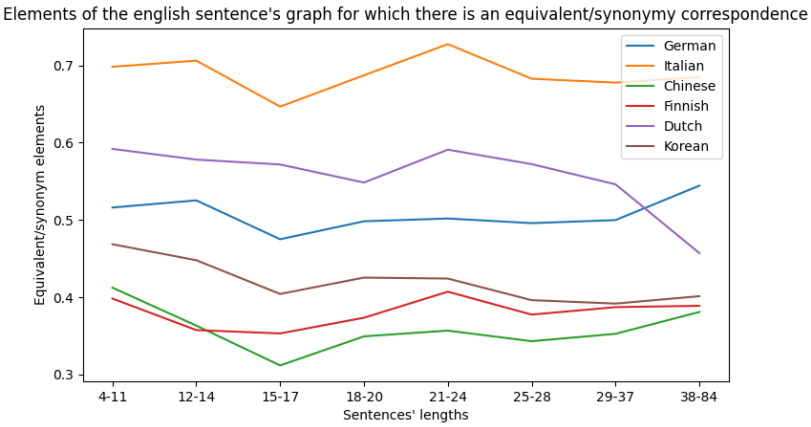

Figure 2.5: Equivalent/syonnymy metric applied to EuroParl + WikiNER dataset

EuroParl Equivalent/synonymy metric average:
IT = 0.659 DE = 0.558 NL = 0.533
ZH = 0.422 KO = 0.411 FI = 0.368

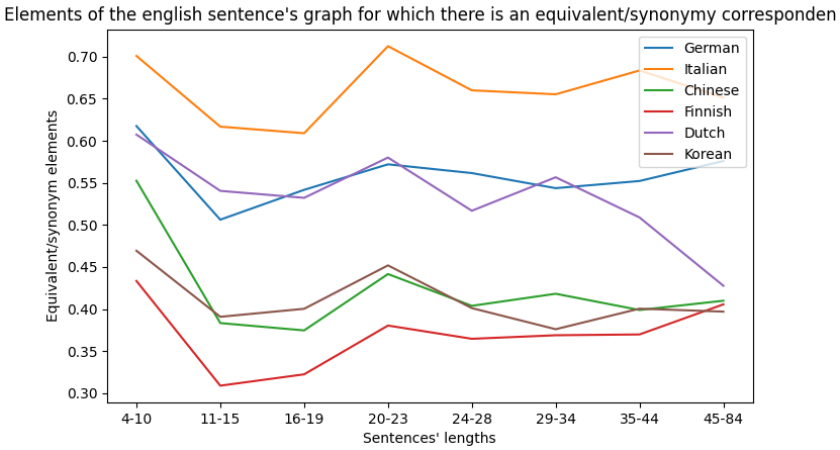

Figure 2.6: Equivalent/syonnymy metric applied to EuroParl dataset

WikiNER Equivalent/synonymy metric average:
IT = 0.714 NL = 0.582 DE = 0.462
KO = 0.430 FI = 0.391 ZH = 0.306

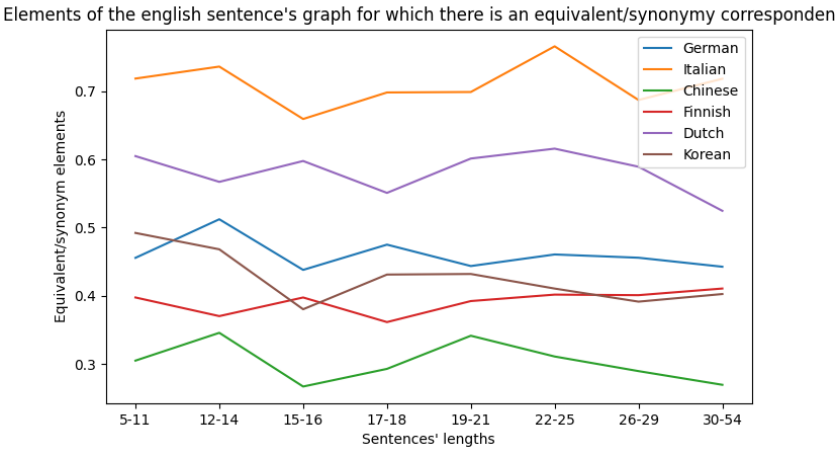

Figure 2.7: Equivalent/syonnymy metric applied to WikiNER dataset

The behaviour of this metric is really similar to the one observed with BLEU score. Indeed the observations done before are valid also here, also respect to the differences between the two sub part of the dataset used (EuroParl and WikiNER).

2.3 OnlyIn metric

The onlyIn metric is based on the predicates onlyIn which indicates those elements of the graphs that the set of rule for the comparison is not able to classify.
Definition:

$$OnlyIn\ score = \frac{number\ of\ onlyIn\ predicates}{total\ number\ of\ predicates} \tag{2.2}$$

OnlyIn metric average:
IT = 0.251 DE = 0.407 NL = 0.362
KO = 0.482 FI = 0.536 ZH = 0.539

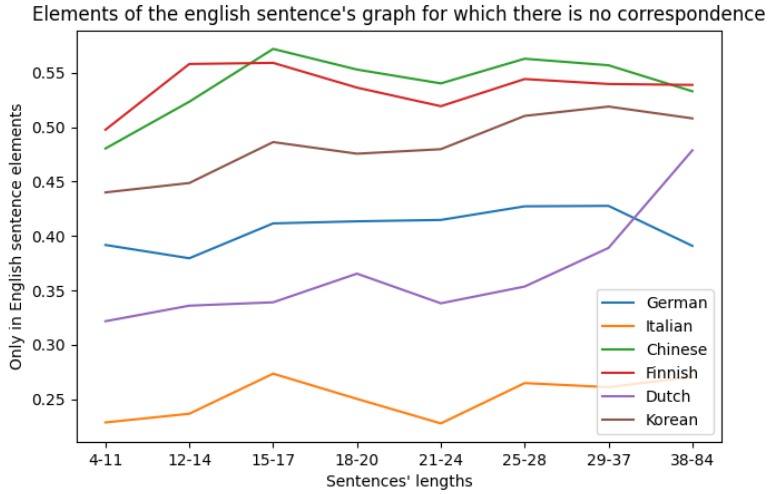

Figure 2.8: OnlyIn metric applied to o EuroParl + WikiNER dataset

The behaviour is exactly the opposite of the precious metrics used. So the higher this score is, the worst the translation.

2.4 Differences between translators

For a really small portion of the EuroParl dataset (155 sentences) I computed their BLEU score for the variations generated with two translators (DeepL and ArgosTranslate).
In this case the number of sentences is pretty low; anyway the behaviours of the two translators respect to the languages and to the sentences' length are pretty different, so I think that this could indicate that the possible causes of those differences may be more technological than linguistic.

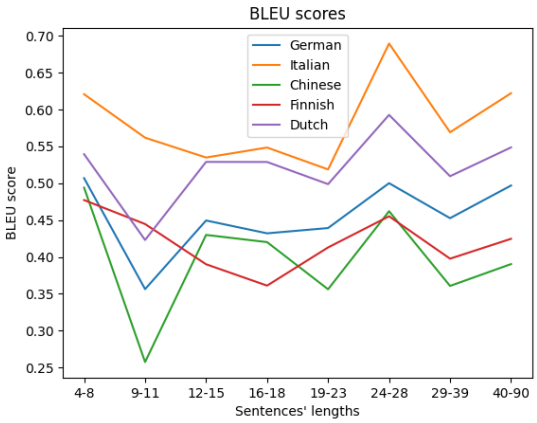

Figure 2.9: BLEU score on DeepL

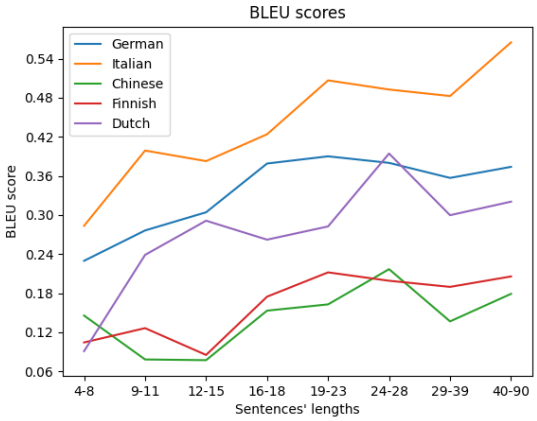

Figure 2.10: BLEU score on Argos Translate

Chapter 3

Semantic types analysis

The last analysis was performed in relation of the semantic types provided by FRED. The analysis take in consideration the classes for which the semantic types is present in the knowledge graphs. I approached this task accomplishing the following steps:

1. extraction of all the elements with one of these prefix:
   - http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#
   - http://www.ontologydesignpatterns.org/ont/d0.owl#
   - http://schema.org/
2. extraction of all the subclasses of the elements retrieved at step 1
3. selection of the triples in the comparison graphs involving the classes retrieved at step 2.

The evaluation is performed with the following metric, similarly as before, but just counting the equivalent predicates:

$$Equivalent\ score = \frac{number\ of\ equivalent\ predicates}{total\ number\ of\ predicates} \tag{3.1}$$

The following graphs shows, using this metric, the coherence of the classes associated with the 11 most frequent semantic types previously retrieved. This is the list of the sematic types:

- http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#Event
- http://www.ontologydesignpatterns.org/ont/d0.owl#Activity
- http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#Situation

- http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#Quality
- http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#InformationEntity
- http://www.ontologydesignpatterns.org/ont/d0.owl#Topic
- http://www.ontologydesignpatterns.org/ont/d0.owl#Location
- http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#PhysicalObject
- http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#Description
- http://www.ontologydesignpatterns.org/ont/d0.owl#Characteristic
- http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#TimeInterval

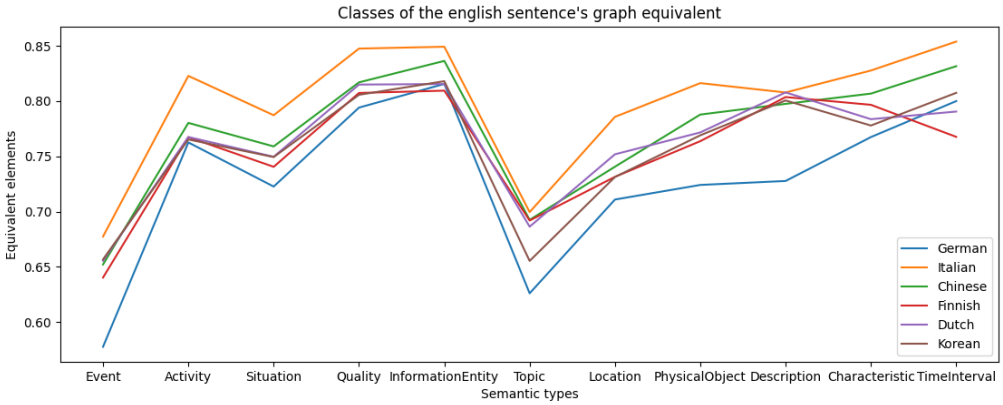

Figure 3.1: Semantic type analysis

Chapter 4

Conclusions

My equivalent/synonymy metric seems to be quite reliable performing similarly to BLEU score, but it's quite inefficient, adding many steps in the middle. Morevover, in those steps language dependant tools have been used. BLEU score, instead, counting just the n-grams of words in common between two sentences is far more efficient and can be used for any language.
Also, there exist some extensions of BLEU score and one of them, called METEOR, extends BLEU integrating the words with stems and synonyms. Our project, deeply relyed on lemmas and synonyms in doing the comparison, so I think that attempts like this, which is to construct a machine translation metric based on knowledge graphs extracted from text, should really go beyond that and perform a deeper semantic analysis in order to provide a real advantage.
For what concern the analysis, I think there isn't a pattern in the scores caused by linguistic characteristics, neither about genealogical relations nor morphological similarities. Indeed, Dutch and German which are genealogical closer to English performed worst than Italian, while Korean (agglutinative, so with a morphology less similar to the English one) don't perform worst than Chinese.
The noticeable differences between the two translators and between the two dataset make me think that the differences on the scores are mainly due to the models used by the translators and the training data used to train them.