Learning Curve with Machine Translation Based on Parallel, Bilingual Corpora

Maciej Kowalski

Abstract Machine Translation is a branch of computer science that automatically handles translation of a text from a source language to a target language. This article summarizes the experience gained during *UKSW* project, part of which deals with translation of legal phrases between English and Polish. The article describes consecutive steps of the project, i.e. collecting data and creating parallel, bilingual corpora, checking open source ready-made solutions and the novel, effective *SMT* solution that has been proposed. The final chapter summarizes the solution, together with the results based on *BLEU* metrics.

Keywords Machine translation · Bilingual corpora · Parallel corpora

1 Background

Machine translation is a part of computer linguistics, which is focused on using automatic methods and computer algorithms in order to translate a text from one language to another. It is nowadays used widely in various areas both professional, like translating technical documents, instruction, reports [1, 2], as well as everyday, such as communicating with other people by using different than mother language,[1] translating pages at run-time, teaching and learning.[2]

Considering both of those applications, many of the current state-of-the-art systems require to be reviewed by human translator who corrects the translation to be

[1] http://translate.google.pl.
[2] http://duolingo.com.

M. Kowalski (✉)
National Information Processing Institute, al. Niepodleglosci. 188B,
00-608 Warsaw, Poland
e-mail: mkowalski@opi.org.pl
URL: http://www.opi.org.pl

D. Ryżko et al. (eds.), *Machine Intelligence and Big Data in Industry*,
Studies in Big Data 19, DOI 10.1007/978-3-319-30315-4_2

more humane. As translation methods and existing solutions are getting better, there is still a need of human interaction and overseeing.

Machine Translation as it is today, may implement one or more of the tree following methods:

- rule-based, also known as knowledge-based translation, which depends on information about two languages having their place in the process. To be able to use this method, one must first analyze both of the languages to find grammar, semantic, morphological or syntactic regularities. After this step, two dictionaries are build for words and rules. Based on the information found, the translation process tries to link the input sentence structure with the output sentence, filling the translation with a data coming from the dictionary.
- example-based (statistical, case-based), which depends on bilingual corpora, to generate statistical model of a language. Given phrase is translated according to probability distribution that an output phrase the translation of it. One of the simplest implementation applies Bayes Theorem.
- hybrid-based, which uses multiple approaches within one machine translation system. The key reason why hybrid approach is being used, is that it has better accuracy than separate approaches.

2 Problem Description

One of the key features of *UKSW* project, was to aid law-related people with translating legal phrases from Polish to English and vice versa. The *UKSW* project consists of several parts, which work together in following way (as an example):

- a person is searching for key phrases, which occur in official court rulings,
- court ruling is described by law acts that are incorporated into it. As a result, user is given not only the text of the ruling, but also all the references to the law acts.
- user can access particular law act and find related court rulings to it.

As it was learned during the *UKSW* project, lawyers or solicitors often make examples or try to find reasoning for their cases in court. In each step of this process, machine translation is required to be working and be available for users, so foreign court ruling and acts may be accessed. That way users are able to find similar cases in other countries.

2.1 Limitations

The *UKSW* project has been limited by two important factors, which are memory size and disk space that the developer/production systems were installed on. Typical machine contained Intel(R) Core(TM) i7-3770 CPU @ 3.40 GHz with 16 GB of

RAM and 300 GB of disk space. Using already proven systems were far more resource-consuming:

- Joshua[3]—RAM consumption depends on amount of parallel data, which the system is trained on. Initial configuration suggest more than 32 GB of RAM.
- Moses[4]—RAM consumption is estimated to be 8 GB for 200,000 items of training data and 100 GB of disk space. Authors suggest that amount of disk space should be 100 times the amount of learning data.

In both cases bottlenecks are RAM or disk space usages, which both excludes those frameworks from being the solution to described problem.

2.2 Data

The translation system described in this article uses parallel bilingual data (Polish and English). The total amount of Polish-English data is approximately 42,000,000 items of words, phrases, sentences and whole documents (different granularity). Data were incorporated from following sources:

- European law acts database—*EUPARL*,[5]
- *TED* talks transcriptions,[6]
- corpus data available at Institute of English Studies at Lodz—*ACADEMIA*, *OSW*, *CORDIS*, *JRC*, *RAPID*, *ESO*, [7]
- dictionaries—*DICT*,[8,9]
- paralell corpora: movies subtitles, medical documentation, manuals, user interfaces of OpenOffice software *DGT*, *EAC*, *ECDC*, *OPUS*, *LIT*,[10]
- eurlex law acts, court orders—*EURLEX*[11]
- European Court of Justice—*CURIA*[12]

For given table (Table 1), $LEN_{min}(EN)$ and $LEN_{max}(EN)$ are the shortest and longest lengths of text blocks found in English part for *source*, $LEN_{min}(PL)$ and $LEN_{max}(PL)$ are the shortest and longest lengths of blocks found in Polish part for *source*. *Count* is the total number of pairs found in *source*.

[3] http://joshua-decoder.org/6.0/pipeline.html.
[4] https://github.com/jladcr/Moses-for-Mere-Mortals.
[5] http://www.statmt.org/europarl/.
[6] http://www.ted.com/talks/browse.
[7] http://pelcra.pl/new/.
[8] http://www.slowniki.org.pl/.
[9] http://www.diki.pl/.
[10] http://opus.lingfil.uu.se/index.php.
[11] http://eur-lex.europa.eu/.
[12] http://curia.europa.eu/.

Table 1 Parallel bilingual data statistics

Source	$LEN_{min}(EN)$	$LEN_{max}(EN)$	$LEN_{min}(PL)$	$LEN_{max}(PL)$	Count
EURLEX	5	1,009,682	0	927,844	531,938
CURIA	0	4595	0	4669	495,924
DGT	31	91,866	30	86,364	2,652,699
EAC	31	1147	31	1123	1699
ECDC	0	1437	31	1512	1970
EUPARL	3	1737	0	1938	629,322
TED	9	98	6	120	3607
CORDIS	2	701	1	934	467,763
RAPID	5	1617	2	1632	371,712
OPUS	2	40,673	5	6871	35,932,031
DICT	7	62	2	201	217,410
ACADEMIA	6	1634	1	1477	17,857
ESO	6	3622	4	3773	4230
JRC	6	4324	0	4448	1,387,708
LIT	6	2210	3	2482	35,434
OSW	6	3718	0	3360	28,718
				Total	42,780,022

Table 2 Size of corpus by granularity

Granularity type	Count
Words	1,024,374
Phrases	14,419,839
Sentences	20,944,836
Documents	6,243,735

For given table (Table 2), words are strings with no white characters included, phrases are strings which have 2–5 white characters included, sentences are strings which have 6–20 white characters included and documents are all strings which contain more than 20 white characters.

All the data was stored using search and indexing engine—*Lucene*.[13] There was no language-specific analyzer used.

Example 1 Structure of *Lucene* document used to store bilingual data record

```
Document {
    <pl:75 OHIM nie zgadza sie z argumentami skarzacej.>
    <en:75 OHIM disputes the applicants arguments.>
}
```

[13] https://lucene.apache.org.

```
Document {
    <pl:77 Nalezy zgodzic sie z ta analiza.>
    <en:77 That analysis must be upheld.>
}
```

3 Process

3.1 Data Alignment

Based on the fact that the data contained entries, which size ranged from few letters up to 1 MB of text, an alignment process should be performed. Standard aligner tools like GIZA++,[14] Berkeley,[15] Hunalign[16] are time and resource consuming processes, therefore an novel alignment method has been introduced that outperforms them.

The indexes between, which result phrase may be located are computed as follows:

$$index_{min} = POS_{SRC}(word)\frac{LEN(block_{SRC})}{LEN(block_{DST})} - \delta \tag{1}$$

$$index_{max} = POS_{SRC}(word)\frac{LEN(block_{SRC})}{LEN(block_{DST})} + \delta \tag{2}$$

For the Eqs. (1) and (2), $POS_{SRC}(word)$ is the position of input phrase inside *source* text block, $LEN(block_{SRC})$ is the length of source text block, $LEN(block_{DST})$ is the length of destination text block, δ is a fixed shift which is by default 2 words.

In the following example, a translation for word "poniewaz" will be searched.

Example Alignment based on the following data block:

```
pl:37. Komisja jest zdania, ze art. 9 ust. 2 lit. dekretu
krolewskiego 14702007 jest sprzeczny z art. 42 ust. 3
rozporzadzenia nr 17822003, poniewaz narusza on zasade
rownosci traktowania.

en:37. The Commission considers that Article 92b of Royal
Decree 14702007 is not compatible with Article 423 of
Regulation No 17822003 because it is inconsistent with
the principle of equal treatment.
```

The word "poniewaz" starts at the index = 135 inside *Polish* part of text block. By using presented formulas (1) and (2), the most probable indexes, between which the translation can be found, are (an assumption is made that $\delta = 2$ words):

[14] http://www.statmt.org/moses/?n=FactoredTraining.RunGIZA.

[15] https://code.google.com/p/berkeleyaligner/.

[16] https://github.com/danielvarga/hunalign.

$$index_{min} = 135\frac{182}{194} - 2_{words} = 106 \tag{3}$$

$$index_{max} = 135\frac{182}{194} + 2_{words} = 139 \tag{4}$$

The most probable space of finding the correct translation is between indexes 106 and 139, which is “regulation No. 17822003 because it is”.

3.2 Translation

General translation method contains following steps:

1. An input phrase is being split into n-grams of the predefined maximum size as follows:

```
Input phrase: A B C D
N-Gram max size: 3
N-Grams: A, B, C, D, A B, B C, C D, A B C, B C D
```

2. Each generated source n-gram is taken as a query to *Lucene* index. If a search has PL-EN direction, the phrase is being searched in PL part of index, otherwise, the EN part is taken into consideration.
3. Each corresponding text block is narrowed down using data alignment method described above.
4. Each narrowed text block found using *Lucene* index is processed by the tokenizer as it is described in point 1. The resulting n-grams are stored in a sorted list, which relates to source n-grams. The more frequent the resulting n-gram is, the higher in the list it is placed.
5. Translation begins from the longest n-grams. The source phrase is, step-by-step being replaced by most probable (the most appearing in results) n-grams.

Example 1:

```
PL: karty do gry

karty -> [cards(114), card(16), you(15), and(10)]
do -> [do(50), to(38), you(22), up(16)]
gry -> [game(79), over(51), games(38), is(14), playing(13)]
karty do -> [playing cards(142), you know them playing cards(140),
            cards and(129), cards of(129)]
do gry -> [back in(38), to play(30), you back in(24), to you(22)]
karty do gry -> [playing cards(166), you know them playing
cards(140), are playing cards it is(132), playing cards and(131)]

EN: playing cards
```

The number given next to the resulting n-gram is the number of coexisting occurrences of n-gram and source n-gram found, using *Lucene* and it is treated as translation score. Given translation result is then cross-checked with more granule translations (appearing higher in list). In following example, "playing cards" is split into two words: "playing" and "cards". If those two words are found in the earlier results, the final score is increased by scores of finer phrases, so in this case final score is $166 + 13 + 114 = 293$ and is the most probable translation. In this example, there exists a phrase, which is translated 1:1 with given input phrase.

Example 2:

```
PL: podejrzany o wlamanie

podejrzany -> suspect(114), is(21), suspicious(14), he(12)
o -> oh(44), it(27), is(19), yeah(15)
wlamanie -> and(50), breaking(46), entering(42), burglary(35)
podejrzany o -> suspect in(146), prime suspect in(127), robbery
suspect(127), suspect on two(122)
o wamanie -> and breaking and entering(109), breaking and
entering(109), breaking and     entering of(104), of breaking
into our(69)
podejrzany o wlamanie -> []

EN: suspect in and breaking and entering
```

In this example there is no 1:1 translation proposition, so the translation must be composed from shorter n-grams. The next n-gram to be analyzed is "and breaking and entering", which is decomposed into words "and", "breaking", "entering". For source n-gram "o wlamianie", this is the most probable translation with score: $109 + 46 + 42 = 192$, which is used as the translation of that part. After that, only "podejrzany" should be translated and the most probable n-gram for this input is suspect with score $= 114$. Overall translation is then composed into one, to be "suspect in and breaking and entering".

4 Variations and Test Results

BLEU is a metric, used for evaluation of quality of machine translated texts in comparison to reference translations. The closer to a reference the machine translation is, the higher the score it has. As an output, metric gives a real number between 0 and 1, for 1 being exact match with given reference and 0 being completely different [3].

The *BLEU* metric is first of its kind, which managed to project the correctness the translated text onto the real number between 0 and 1. Due to the simplicity of implementation and speed of operation, it is still used to measure the quality of machine translated texts. Presented method was assessed by using *BLEU*.

Table 3 Translation test cases (including common and legal phrases)

English common phrase	Polish common phrase
I want to press the point	chce odkreslic, ze
Willingly	chetnie
What I meant is	chodzi mi o to, ze
At times	chwilami
To the moment	co do sekundy
English legal phrase	Polish legal phrase
Criminal action	proces karny
Criminal proceedings	postepowanie karne
Death tax	podatek od spadku
Debt collector	windykator
Default judgment	wyrok zaoczny

Table 4 BLEU score for common phrases

	ORIGINAL	FIXED	MORFOLOGIK	EXTENDED
COMMON	0.48	0.48	0.22	**0.52**
LEX	0.31	0.32	0.20	**0.40**

During experiments, the following variations of this method were tested:

- as described, with n-gram result list = 8—*ORIGINAl*,
- during analysis of the data, based on which the translation is being made, author found data errors, which may influence the overall score. The text errors were related to wrong UTF-8 character encoding, which gave wrong text for Latin characters in Eurlex corpus—*FIXED*,
- with using *Morfologik*[17] analyzer polish data—*MORFOLOGIK*,
- with extended with n-gram result list = 20—*EXTENDED*.

For testing purposes, two sets of 100 pairs of Polish-English phrases were created (Table 3):

- concerning every-day language—*COMMON*,
- concerning legal phrases—*LEX*.

For translation from Polish to English, the test gave following *BLEU* results (Table 4):

For translation from English to Polish, the test gave following *BLEU* results (Table 5):

There is a percentile of translations, which were measured incorrectly. This happens because translation reference does not take into consideration variations of

[17] http://morfologik.blogspot.com.

Table 5 BLEU score for legal phrases

	ORIGINAL	FIXED	MORFOLOGIK	EXTENDED
COMMON	0.33	0.33	0.31	**0.39**
LEX	0.21	0.22	0.21	**0.25**

Table 6 Example translations

Phrase	Translation	Reference translation
w zwiazku z	has to do with	with reference to
co do sekundy	to the second	to the moment
blisko lez	close tears	near tears
dobrze zgadnac	okay guess	guess right
moim zdaniem	in my opinion	to my mind

given phrase (The same phrase may be translated in different way, dependent on the context). Examples of false-negative translations (Table 6):

5 Application

As an end result, a web service (*REST*), which implement above process, has been created. The use of this web service is as follows:

Usage:

```
http://localhost:8080/translate-web/rs/translate/{INDEX}?
                      direction={DIRECTION}&phrase={PHRASE}

INDEX - selected lucene index [COMMON | LEX]
DIRECTION - translation direction [ PL_EN | EN_PL ]
PHRASE - input phrase to translate
```

Example:

```
http://localhost:8080/translate-web/rs/translate/COMMON?
                      direction=PL_EN&phrase=karty
```

As result, a *JSON* object is produced. It holds each step of the translation process, which with the help of user's knowledge, provides useful translation help.

Result:


```
{
   "translations":[
      {"translation":"karty","comment":"cards, and, card, you"},
      {"translation":"do","comment":"do, to, you, on"},
      {"translation":"gry","comment":"game, games, over, of"},
      {"translation":"karty do","comment":"playing cards"},
      {"translation":"do gry","comment":"back in, to play"},
      {"translation":"karty do gry","comment":"playing cards"}],
   "best":"playing cards",
   "valid":true,
   "message":null
 }
```


6 Conclusions

Presented solution cannot compete against currently working *SMT* solutions like Joshua and Moses. As it is described in Moses manual [5] for corpora of 200,000 pairs, system has been able to produce the result, which was scored up to 0.78 using *BLEU* metric. Similar example [4] based on *Euparl* corpora alone, produced up to 0.20 higher than the described solution. Although the simplicity of the process and small amount of resources necessary to aid a user to translate a phrase makes this approach legible and useful. Following data set has been prepared with on machine with configuration described in Sect. 2.1 in this chapter. Generated *Lucene* index consumed around 10 GB of disk space. The building process for around 42,000,000 pairs of text blocks took 3 h and 32 min. Approach which uses Moses [5] uses ten times more disk space and lasts up to 20 h for a model build. In order to improve the score, author suggest implementing simple language model for phrases. Therefore it is necessary split and align the longest text blocks used in this process.

References

1. Bond, F.: Machine translation introduction - lecture 1. NTT Communication Science Laboratories (2006)
2. Arnold, D.J., Balkan, L., Meijer, S., Humphreys, R.L., Sadler, L.: Machine Translation: An Introductory Guide. Blackwells-NCC, London (1994)
3. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, pp. 311–318 (2002)

4. Koehn, P.: Europarl: a parallel corpus for statistical machine translation. In: MT Summit, vol. 5, pp. 79–86 (2005)
5. Machado, J.M., Hilario, L.F.: Moses for Mere Mortals. Tutorial. A machine translation chain for the real world (2014). https://github.com/jladcr/Moses-for-Mere-Mortals/blob/master/Tutorial.pdf

http://www.springer.com/978-3-319-30314-7

Machine Intelligence and Big Data in Industry

Ryżko, D.; Gawrysiak, P.; Kryszkiewicz, M.; Rybiński, H. (Eds.)

2016, VIII, 236 p. 62 illus., 39 illus. in color., Hardcover
ISBN: 978-3-319-30314-7