TER-TIAD 2021
Translation inference across dictionaries

THAI Ba Tuan

Supervisor: Carlos Ramisch

Outline

1. Introduction
2. Data Processing
3. Algorithms
4. Experiments and Results
5. Conclusions and Future works

Outline

1. Introduction
2. Data Processing
3. Algorithms
4. Experiments and Results
5. Conclusions and Future works

Introduction

- The fourth shared task for Translation Inference Across Dictionaries (TIAD 2021)
- Goal is to generate new translation pairs (bilingual dictionaries) based on provided bilingual dictionaries, BUT the existing dictionaries do not contain the target language pairs. Add an example.
- TIAD-2021 will be held In Zaragoza (Spain) on September 1, 2021
- TIAD-2021: English(EN), French(FR), Portuguese (PT)
- Generate 6 dictionaries (EN-FR, EN-PT, FR-EN, FR-PT, PT-EN, PT-FR)
- Participants may also make use of **other freely available sources of background knowledge** (e.g. lexical linked open data and parallel corpora) to improve performance.
- No direct translation among the target language pairs is applied.
- Submit dictionaries on 14/05/2021 (have been submited)
- Result announcement 14/06/2021

Outline

1. Introduction
2. **Data Processing**
3. Algorithms
4. Experiments and Results
5. Conclusions and Future works

2.1 Data source

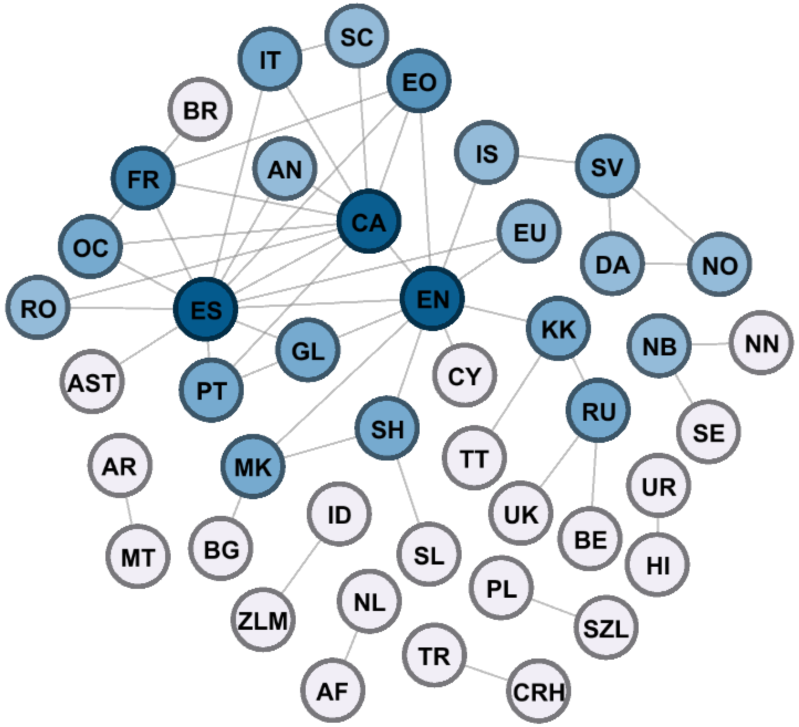
- Apertium relies on a set of bilingual dictionaries, developed by a community of contributors, which covers more than 40 languages pairs.
- Apertium RDF is the result of publishing the Apertium bilingual dictionaries as linked data on the Web, It contains 44 languages and 53 language pairs
- How to get the data source:

Option 1

- Apertium RDF v2, SPARQL query
- *Ontolex lemon* core model
- 44 languages and 53 language pairs
- Use SPARQL query to get direct translations

Option 2

- CSV “shortcut”
- 51/53 language pairs csv files
- Information:
  - source written, source lexical (URI), source sense (URI),
  - translation (URI)
  - target sense (URI), target lexical (URI), target written
  - part of speech (URI)



➢ **Choose option 2 for saving time and easy to process**

2.2 Data preprocessing

	"written_rep_a"	"lex_entry_a"	"sense_a"	"trans"	"sense_b"	"lex_entry_b"	"written_rep_b"	"POS"
0	"waiting for"	"http://linguistic....r-ADP-en"	"http://linguistic....en-sense	"http://linguistic....se-	"http://linguistic....es-	"http://linguistic....a-ADP-es"	"esperando a"	"http://www.lexinfo...position"
1	"split"	"http://linguistic....-VERB-en"	"http://linguistic....en-sense"	"http://linguistic....se-	"http://linguistic....es-	"http://linguistic....-VERB-es"	"partir"	"http://www.lexinfo...nfo#verb"
2	"Donostia-San Sebastián"	"http://linguistic....PROPN-en"	"http://linguistic....en-sense"	"http://linguistic....se-	"http://linguistic....es-	"http://linguistic....PROPN-es"	"Donostia-San Sebastián"	"http://www.lexinfo...operNoun"
3	"little"	"http://linguistic....e-ADV-en"	"http://linguistic....en-sense"	"http://linguistic....se-	"http://linguistic....es-	"http://linguistic....o-ADV-es"	"poco"	"http://www.lexinfo...o#adverb"
4	"send back"	"http://linguistic....-VERB-en"	"http://linguistic....en-sense"	"http://linguistic....se-	"http://linguistic....es-	"http://linguistic....-VERB-es"	"devolver"	"http://www.lexinfo...nfo#verb"

	source	target	POS
0	waiting for	esperando a	preposition
1	split	partir	verb
2	Donostia-San Sebastián	Donostia-San Sebastián	properNoun
3	little	poco	adverb
4	send back	devolver	verb

2.2 Data preprocessing

- The Dictionary from English to Welch (EN – CY) or

from English to Kazakh (EN – KK) can remove

- Because: No inference path from English, French, Portuguese use KK or CY

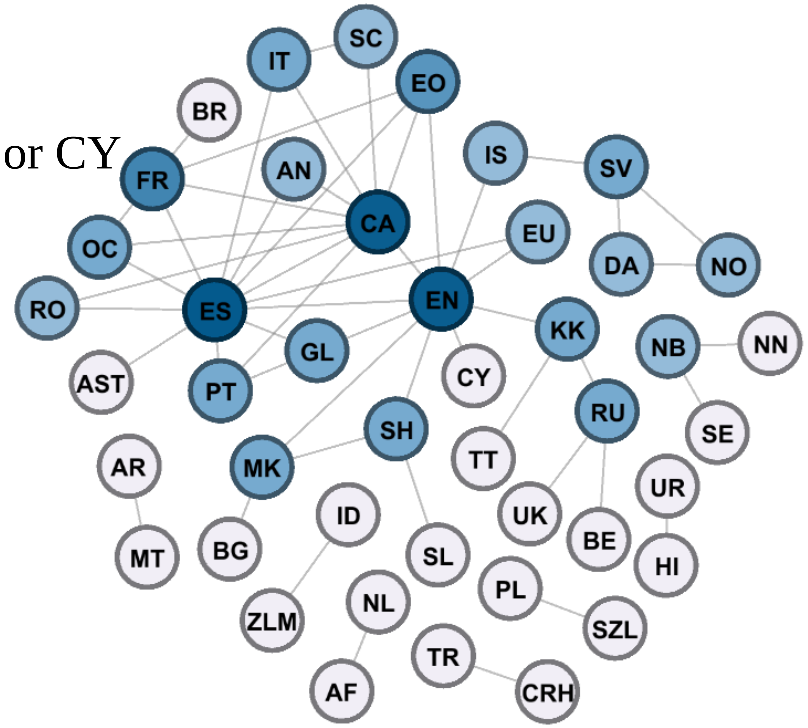
as a pivot

How to list all related dictionaries?

- Using Depth First Search to find all possible paths from:

EN- FR, EN-PT, FR-PT

=> 27 bilingual dictionaries to build inference dictionaries



2.3 Statistic Data

Table 1 Number of POS for each lang and total

	all	EN	PT	FR	CA	ES
noun	307,599 (43.3%)	26,707	11,475	27,798	33,050	35,829
properNoun	166,166 (23.39%)	43,856	24,210	36,626	42,573	34,225
adjective	107,067 (15.07%)	9990	9,670	11,519	17,100	19,939
verb	70,248 (9.89%)	6640	2,519	6,208	6,574	9,015
adverb	44,566 (6,27%)	4756	2,556	3,670	8,010	9,556
Total	710,441	93,463	51,035	87,145	108,646	110,447
Total records						

Table 2 Number of Inference word for each langague

	EN	PT	FR
properNoun	20,978 (35.2%)	24,210	36,626
noun	19,945 (33.5%)	11,475	27,798
adjective	7996 (13.42%(	9,670	11,519
verb	5,299 (8.9%)	2,519	6,208
adverb	4,266 (7.16%)	2,556	3,670
Total	59,576	82,200	51,035

- Table 1 show all word (remove duplicates) of data in the TIAD-2021 and some langague

Because some words can appear many times (ex: life noun)

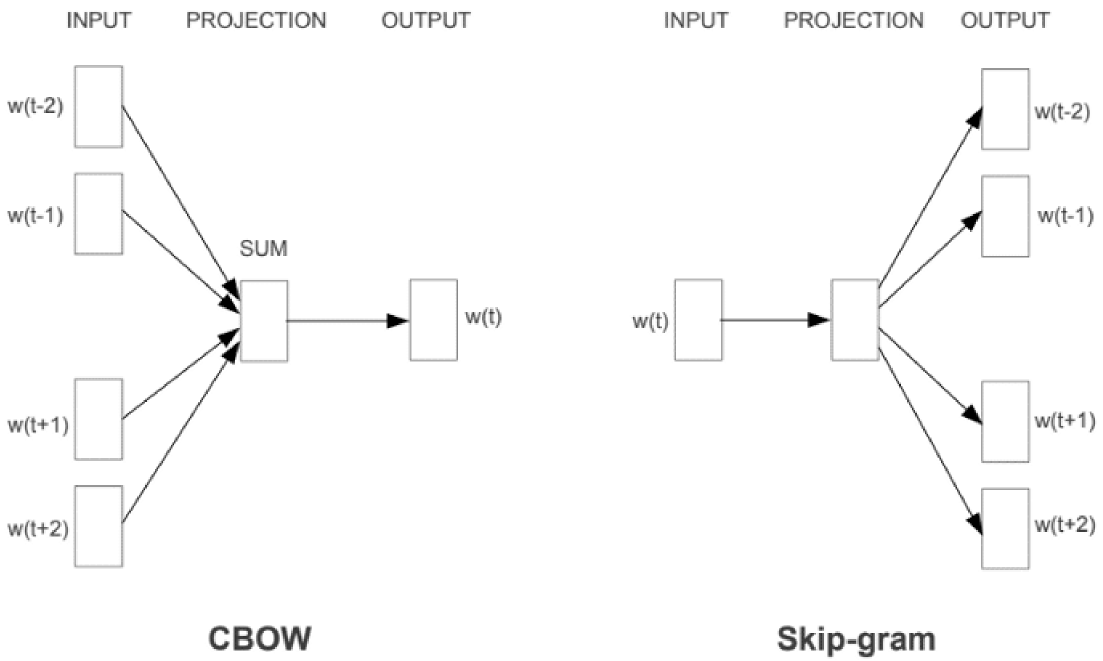
- Table 2 show the word need to inference of each langague after filter by DFS

Outline

1. Introduction
2. Data Processing
3. **Algorithms**
4. Experiments and Results
5. Conclusions and Future works

3.1 Word2Vec

- Introduced in 2013 by Mikolov et al.
- Statistical method for learning a standalone words embedding from text corpus
- Feed-forward, fully connected neural network
- There are 2 methods:
  - Bag-of-words (CBOW): learn from its context
  - Skip-Gram: predict surrounding context word



Source: Mikolov et al. 2013a

3.1.1 Implementation Word2Vec

- With a dictionary lang1-lang2 create pseudo-sentence:

X_lang1_pos Y_lang2_pos X_lang1_pos Y_lang2_pos

for each translation in this dictionary

Example: dictionary EN-ES:

  - life vida noun → life_EN_noun vida_ES_noun life_EN_noun vida_ES_noun
  - GPS GPS noun → GPS_EN_noun GPS_ES_noun GPS_EN_noun GPS_ES_noun

- Build a single **corpus** from 27 bilingual dictionaries => corpus for learning word embeddings
- Why it works?
  - If use window size is 1
  - The word "vida_ES_noun" is predict by the word "life_EN_noun" and vice versa.
  - If we have a translation from FR-ES: "vie vida noun" -> vie_FR_noun vida_ES_noun vie_FR_noun vida_ES_noun.

⇒ Translation from "life noun" from EN -> FR is "life vie noun"

3.1.2 Building dictionary

- Using *Cosine similarity:*
  $cos\,\theta = \frac{A*B}{||A||*||B||}$
- Building dictionary for 6 pairs

langagues

Algorithm 1: Building Inference dictionary Word2Vec
Input: source lang, target lang, word embedding model
Output: dictionary of two languages

1. Pick all the word of source and target language in Word2Vec model by get all the word which have "_source-lang_" or "_target-lang". They are W1 and W2
2. For each word in W1:
   1. Calculate score cosine similarity of this word to all word in W2
   2. Order score descending
   3. Get the first word with the same pos with entry word
   4. Make the output: "source word \t target word \t pos \t score"
      Where score is the confidence of this translate
3. Return Dictionary

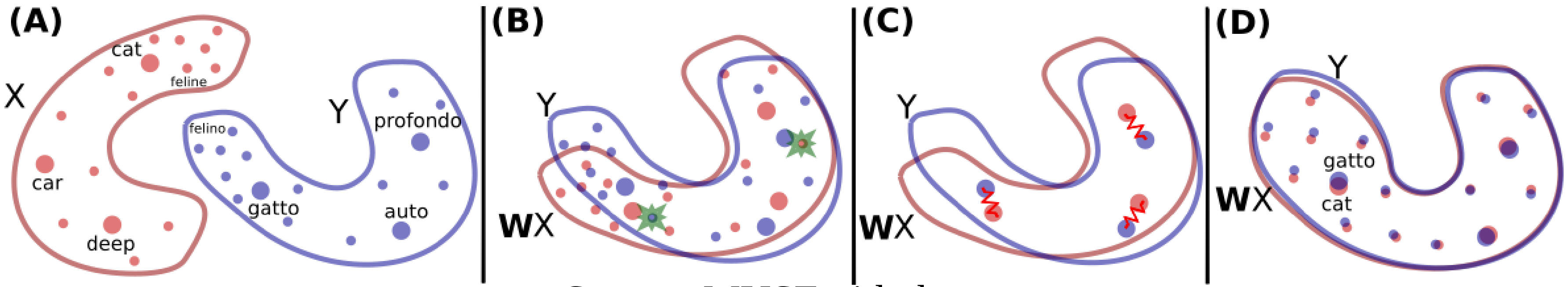
Note: By the request of the organizers
All dictionaries submit have format TSV (tab-separated)
Contains: "source" "target" "Part of speech" "confidence score"

3.2 Cross-lingual word embedding

- Cross-lingual word embeddings simply refers to word embeddings in two or more languages that are aligned to a common space
- MUSE: Multilingual Unsupervised and Supervised Embeddings, Intro by Facebook AI 2018
- State-of-the-art multilingual word embeddings

Find mapping matrix from lang1 to lang2 by optimize $W^* = argmin_{W \in dxd} ||WX - Y||$

- X, Y is dictionary of n pairs of word (n=5000, 10000, …)
- Translation t of any source word s is $t = \text{argmax}_t \cos(Wx_s, y_t)$



Source: MUSE github

3.2.1 Implementation MUSE

- Create mapping matrix from Source-lang to Target-lang (ex: EN -> FR)
- No direct dictionaris => choose Pivot-Lang
- Learning mapping matrix from Source->Pivot, and Pivot -> Target
- Product two mapping matrix

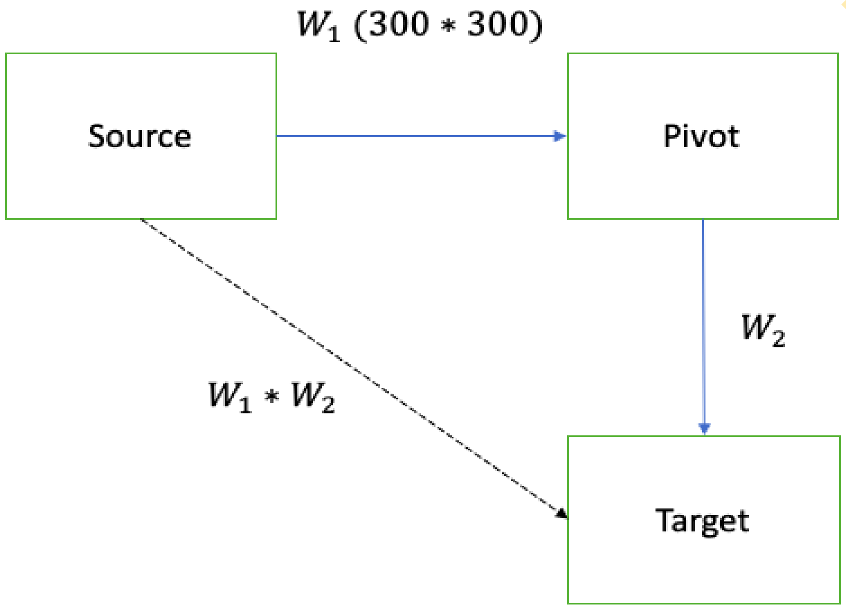
Why it works?

For a word in Source-lang representation with vector **x:**

- $x \rightarrow x\, W_1 \sim y$ **in the pivot-lang**
- and $y \rightarrow y\, W_2 \sim z$ in the target-lang
- With source-lang vector $x \rightarrow x\, W_1 W_2 \sim yW_2 \sim z$ in the target-lang

We can find a target word embeeding with similiraty for the source word

- Using Supervised method (Using dictionary TIAD)



3.2.2 Building Dictionary

Algorithm 2: Building Inference dictionary from MUSE
Input: source-words, target-words, source embeddings, target embeddings, mapping matrix source-pivot W1, mapping matrix pivot-target W2
Output: dictionary of two languages

1. Calculate the mapping matrix from source to target $\boldsymbol{W_1 W_2}$
2. For each word $\boldsymbol{x}$ in source-words:
   1. Get the vector representation of x is v
   2. Calculate vW_1W_2 is v1
   3. Calculate cosine similarity score of v1 to all vector in target embeddings
   4. Sort the score and get the word with the same pos with input
   5. Make the output: source word \t target word \t pos \t score
      Where score is the confidence of this translate
1. Return Dictionary

Note:
The dictionary to submit have format TSV (tab-separated)
Contain: ”source” ”target” “Part of speech” ”confidence score”

3.2 Systems

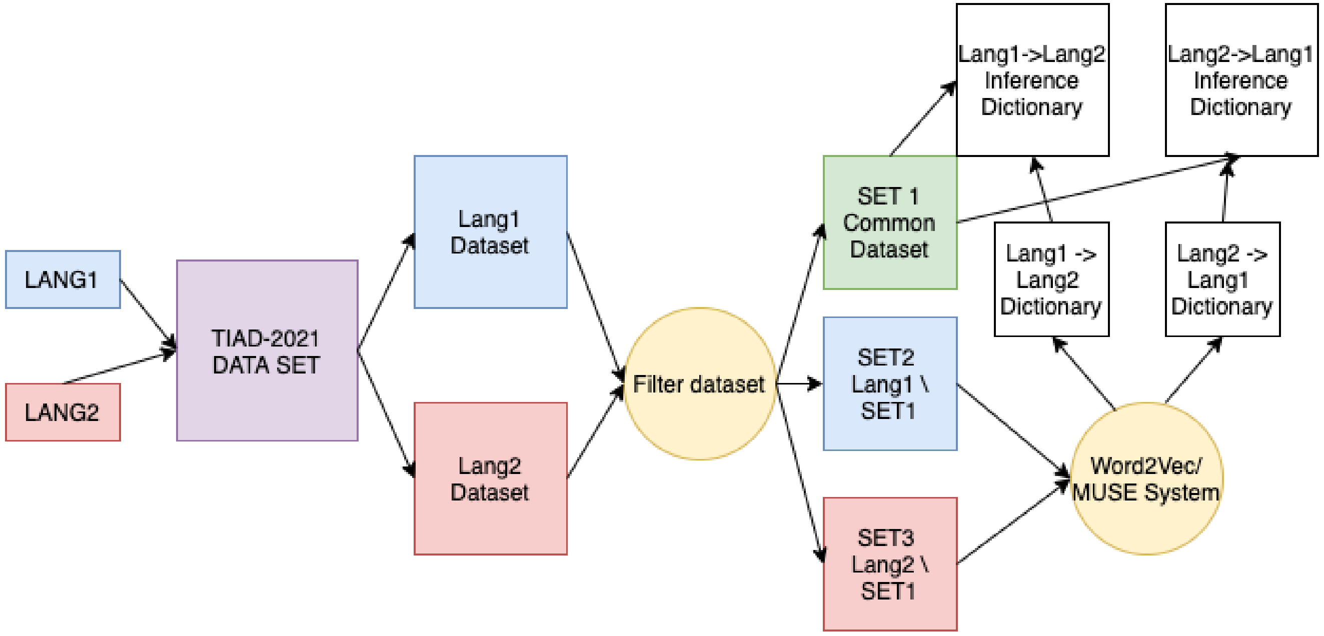
- List all word common in Source-Target (which have same word, same pos)

ex: Brook Brook properNoun (EN-FR)

- Find the inference for the other

Pair	Size
EN-FR	20738
FR-PT	13373
PT-EN	23219

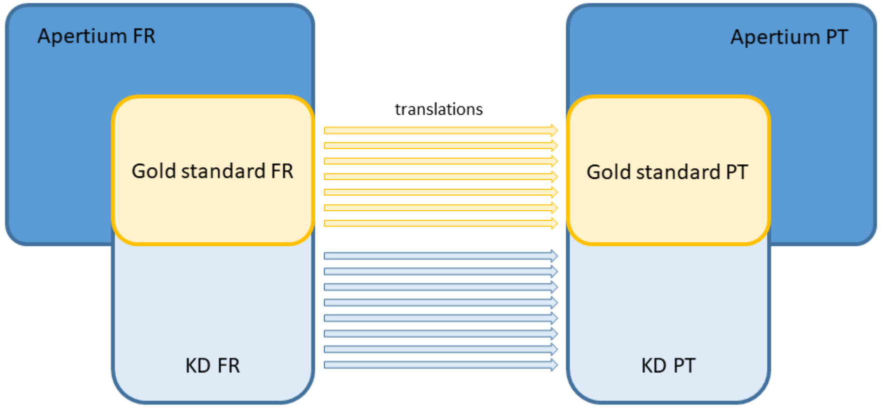
Table 3 List of all common word for each pair



Outline

1. Introduction
2. Data Processing
3. Algorithms
4. **Experiments and Results**
5. Conclution and Future works

4.1.1 Data Evaluation

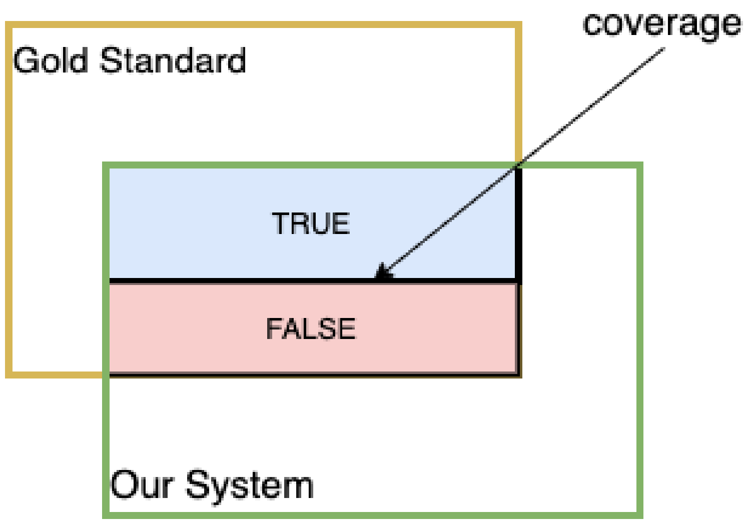


Source TIAD

- Extract Gold Standard (Using KDictionaries)
- Calculate by F-measure
- Find Intersection beetween Gold Standard and Systems

$$Precision(P) = \frac{|TRUE|}{|TRUE|+|FALSE|} \quad Recall(R) = \frac{|TRUE|}{|Gold\ Standard|}$$

$$F - measure = 2/(\frac{1}{R} + \frac{1}{P}) = \frac{2 * R * P}{R + P}$$



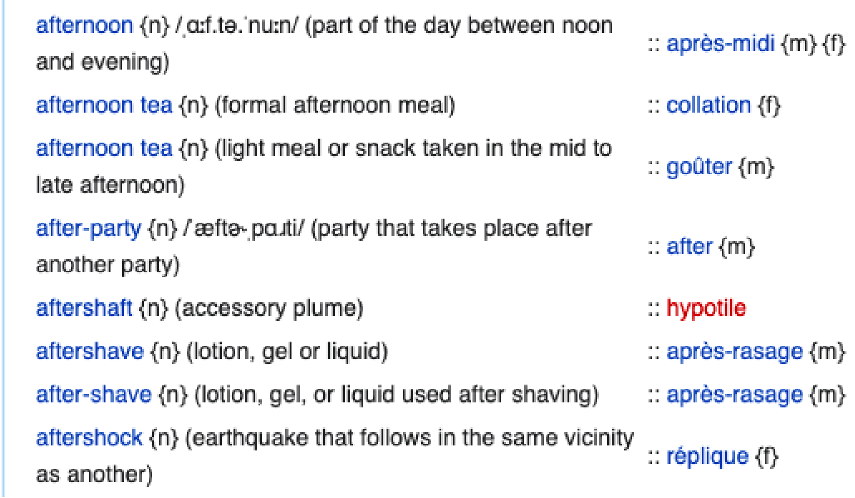
- We can not access to Kdictionaries (limited 100-queries for day)

=> Building Data

4.1.2 Builing Data evaluation

Method 1

- Wikitionary dumps
- Scraping from Matthias Buchmeier generated dictionary
- 6 dictionaries



afternoon {n} /ˌɑːf.tə.ˈnuːn/ (part of the day between noon and evening) :: après-midi {m} {f}
afternoon tea {n} (formal afternoon meal) :: collation {f}
afternoon tea {n} (light meal or snack taken in the mid to late afternoon) :: goûter {m}
after-party {n} /æftə-pɑɹti/ (party that takes place after another party) :: after {m}
aftershaft {n} (accessory plume) :: hypotile
aftershave {n} (lotion, gel or liquid) :: après-rasage {m}
after-shave {n} (lotion, gel, or liquid used after shaving) :: après-rasage {m}
aftershock {n} (earthquake that follows in the same vicinity as another) :: réplique {f}

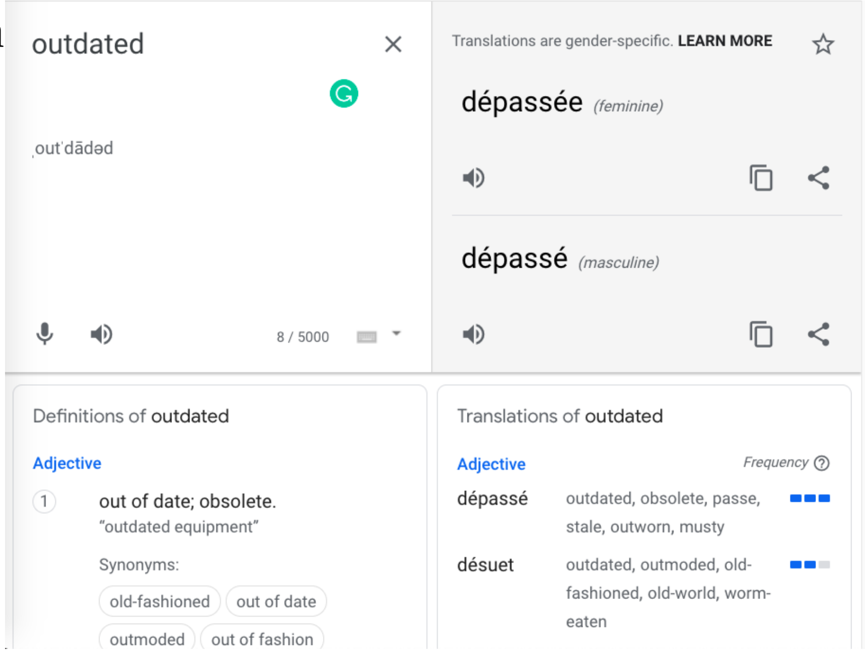
Example EN-FR from Matthias Buchmeier

Method 2

- Google Translated
- Avoid missing word
- Using selenium (python library)
- Extract POS, translation
- Using Xpath

Table: Size of dictionary

Lang, Pair	Size (Wiki)	Size (Google Translate)
EN-FR	118,272	159,104
EN-PT	102,269	103,597
FR-EN	125,945	68,584
FR-PT	29,340	74851
PT-EN	91,813	120,399
PT-FR	12,235	49,517



4.2 Word2Vec

- Gensim Library
- Building corpus from 27 csv files:
  - 66.6 MB text and 810598 lines
- Configuration:
  - Skip-Gram and Cbow, 200 epochs,
  - vector size 300, window size = 1
  - min count = 2 (to get all vocabularies)
- Result: Correct EN->FR

apple	pomme	noun	0.99
approximate	approximatif	adjective	0.95
arm	bras	noun	0.93

- Wrong

in practice	effrénément	adverb	0.56
in reply	à bout de souffle	adverb	0.5
20th Century Fox	Gemeaux	properNoun	0.51

Table results using Cbow

	Wiktionary			Google Translate			Accuracy	
	Precision	Recall	F-measure	Precision	Recall	F-measure	Wiktionary	Google translate
EN-FR	0.7606	0.3304	0.4607	0.5185	0.1898	0.2779	0.3325	0.4966
EN-PT	0.7604	0.3153	0.4458	0.3372	0.1112	0.1673	0.3077	0.1915
FR-EN	0.4502	0.2526	0.3236	0.5001	0.2009	0.2866	0.1877	0.165
PT-FR	0.7519	0.666	0.7064	0.4146	0.4274	0.4209	0.1024	0.4146

Table results using Skip-gram

	Wikipedia			Google Translate			Accuracy	
	Precision	Recall	F-measure	Precision	Recall	F-measure	Wiktionary	Google translate
EN-FR	0.7592	0.3298	0.4598	0.5168	0.1892	0.2770	0.3318	0.4951
EN-PT	0.7615	0.3157	0.4464	0.3361	0.1109	0.1667	0.3081	0.1909
PT-EN	0.5932	0.3134	0.4101	0.5482	0.2351	0.3291	0.1971	0.5477
PT-FR	0.7481	0.6625	0.7027	0.4137	0.4264	0.42	0.1018	0.4136

4.3 MUSE

- Pivot lang: Catalan (CA), Spanish (ES) – which have more connections
- Learning fastText word embeedings for each Langague using data from CoNLL 2017 Shared Task
- CoNLL-U Format: ID, FORM, LEMMA, UPOS,…
- Extract word with LEMMA and UPOS for each lang and build new sentence:

  Ex: thank_verb you_ pronoun for_adposition your_ pronoun time_noun ._punctuation

```
# sent_id = email-enronsent19_02-0049
# text = Thank you for your time.
1       Thank   thank   VERB    VBP     Mood=Ind|Tense=Pres|VerbForm=Fin        0       root    0:root  _
2       you     you     PRON    PRP     Case=Acc|Person=2|PronType=Prs  1       obj     1:obj   _
3       for     for     ADP     IN      _       5       case    5:case  _
4       your    you     PRON    PRP$    Person=2|Poss=Yes|PronType=Prs  5       nmod:poss       5:nmod:poss     _
5       time    time    NOUN    NN      Number=Sing     1       obl     1:obl:for       SpaceAfter=No
6       .       .       PUNCT   .       _       1       punct   1:punct _
```

- Building corpus for English(EN), French(FR), Portuguese (PT), Catalan (CA), Spanish (ES)
- About 4.5GB text for each corpus
- Learning word Embeedings with fastText: vector size 300, min count 4, epochs 15, window size 4, loss hs

4.3 MUSE

- Buiding dictionary for each pair:
- Example: EN -> ES -> FR

little_adverb	poco_adverb
metallic_adjective	metálico_adjective
retailer_noun	detallista_noun
by-train_adverb	en-tren_adverb

EN-ES

decano_noun	doyen_noun
reformación_noun	réformation_noun
paraná_propernoun	parana_propernoun
aleatorio_adjective	aléatoire_adjective

ES-FR

- Similar for: EN->ES->PT, EN-CA-FR, ….
- Learning mapping matrix
- Configuarion: epochs15, cuda False

4.3 MUSE

- Generally, MUSE not good as Word2Vec
- Take long time to build dictionary
  - 1s / 1 translation (11 hours to build English-French)
  - Not have Portuguese -> English use
    Catalan as pivot, missing some words
- Can find the translate of word which not

appear in TIAD dataset:

Example: English to French dictionary:

diversified diversifié adjective 0.4661188

- Not handel the uppercase word (MUSE source

code not allow)

Table: Using ES pivot

	Wikipedia			**Google Translate**			**Accuracy**	
	Precision	Recall	F-measure	Precision	Recall	F-measure	Wiktionary	Google translate
EN-FR	0.8714	0.3143	0.4619	0.6631	0.1314	0.2194	0.3772	0.6531
EN-PT	0.1316	0.0424	0.0642	0.1609	0.0234	0.0409	0.0563	0.0958
FR-EN	0.1558	0.0801	0.1058	0.1892	0.035	0.0591	0.0698	0.0552
FR-PT	0.0632	0.0363	0.0461	0.1879	0.0842	0.1163	0.0095	0.1709
PT-EN	0.1338	0.0683	0.0904	0.4148	0.0841	0.1398	0.0387	0.4141
PT-FR	0.0407	0.0356	0.038	0.2569	0.2314	0.2435	0.006	0.2569

Table: Using CA pivot

	Wikipedia			**Google Translate**			**Accuracy**	
	Precision	Recall	F-measure	Precision	Recall	F-measure	Wiktionary	Google translate
EN-FR	0.3938	0.1379	0.2043	0.5154	0.0949	0.1603	0.1592	0.507
EN-PT	0.5987	0.2344	0.3369	0.2977	0.0679	0.1106	0.2793	0.169
FR-EN	0.2182	0.1161	0.1516	0.2597	0.035	0.0617	0.0853	0.0552
FR-PT	0.067	0.0384	0.0488	0.1919	0.0842	0.1171	0.0097	0.0673
PT-EN	no file	no file	nofile	no file	no file	nofile	nofile	nofile
PT-FR	0.0402	0.0352	0.0376	0.2554	0.2314	0.2428	0.0059	0.2553

Outline

1. Introduction
2. Data Processing
3. Algorithms
4. Experiments and Results
5. **Conclusions and Future works**

5. Conclusions and Future works

- With this project, I have learned a lot of knowledge:
  - Word2Embeedings and is application
  - Cross-lingual word embedding with MUSE
  - Processing data with Pandas library
  - Web scraping with selenium, beautiful soup
  - Practice programming with Python, using bash scripts, using source version control with Github
  - Reading paper and writing report

Future works:

- Improve MUSE, find the other method (try implementation in TIAD-2022)
- Consider the “sense” of word

Github repository: https://github.com/batuan/TER_TIAD_2021

Questions