Content

Introduction

Research question

Can we use vector representation of words to detect the difference between inflection and derivation?

Methodology

This study explores the distinction between inflection and derivation in Polish and Spanish using various word embedding models, both static and contextual.

To achieve this, we implemented several models. For static embeddings, we used Word2Vec and FastText, and for contextual embeddings, we utilized the Multilingual BERT model. The Word2Vec model for Spanish was based on the Spanish Billion Words corpus. For Polish, we used the IPIPAN Word2Vec model (nkjp+wiki-forms-all-300-skipg-ns), which is comparable

in quality to SBW. It was trained on the National Corpus of Polish (NKJP) and Wikipedia, includes all parts of speech and word forms, and produces 300-dimensional vectors using the skip-gram algorithm with negative sampling.

In addition, we applied FastText embeddings for both Spanish and Polish to incorporate subword-level information. For contextual representation, we used Multilingual BERT (mBERT) to capture context-sensitive nuances in the morphological structures.

In order to conduct an initial analysis we constructed two separate datasets, using data from UniMorph, one for inflection and another one for derivation.

Initial analysis

Datasets

For the inflection analysis, we constructed a Pivot/Inflection dataset limited to verb-to-verb (V:V) transformations, since no other category is possible.

- We filtered the data to include the following verb tenses in Spanish:
  - Present Indicative. UniMorph category: V;IND;PRS.
  - Past Imperfect. UniMorph category: V;IND;PST;IPFV
  - Future Indicative. UniMorph category: V;IND;FUT

This resulted in a dataset of 148,051 rows, each consisting of a base form, its inflected variant, and the morphological category. Additional forms such as participles and gerunds are planned for future inclusion.

- In Polish the filtering included:
  - Present. UniMorph category: V;PRS.
  - Past. UniMorph category:V;PST.
  - Future. UniMorph category:V;FUT.

Table 1: Mean similarity in initial results

(a) Mean similarity in inflection

Model	Language	Initial Analysis
FastText	Spanish	0.511
	Polish	0.486
Word2Vec	Spanish	0.504
	Polish	0.513
Mult BERT	Spanish	0.927
	Polish	0.910

(b) Mean similarity in derivation.

Model	Language	Initial Analysis
FastText	Spanish	0.511
	Polish	0.544
Word2Vec	Spanish	0.504
	Polish	0.406
Mult BERT	Spanish	0.927
	Polish	0.931

The resulting dataset contains 23,615 rows, structured similarly to the Spanish set with base, inflected form, and category.

For the derivation analysis the data provided by UniMorph was used without changes.

Initial results

The initial results are shown in the following tables.

Now we will plot the initial results. Mean similarity of the category (V:V, ADJ:N, ADV:ADJ…) by type (inflection, derivation between the same categories and derivation between different categories), separated by embeddings model and language.

We can see in **?@fig-initial** some weird outliers in FastText and Word2Vec. These outliers correspond to different instances of U:U, X:U or U:X (X being any label). This unknown category is messing with the means so the data needs to be cleaned a bit and we need to keep N, ADJ, ADV and V.

Cleaning the datasets

Cleaning derivations

The first analysis revealed some errors in both datasets. In the derivations dataset the label U (that *possibly* means unspecified or unknown) presented some issues. The goal is to eliminate

all instances of unknown categories, to get rid of this noise and have cleaner results and means.

1. In Spanish there are 20 rows that contain a derivation that results in U (i.e. N:U or V:U) and 107 in Polish.
2. On the other hand, there are even more derivations in which the pivot is tagged with U (U:N, U:ADJ…), 36 in Spanish and 253 in Polish.

Taking a quick look through this data one can see many mistakes such as clear verbs, adjectives or nouns being labeled U and also formatting issues. When it comes to Spanish, the number is not too high, so it is something that is worth fixing in order to get rid of this label so it does not mess with the means shown in **?@fig-initial**. Fixing the first group seems fairly easy since we can just look at the affix and assign the category it gets assigned in other instances.

Spanish data

In order to clean the Spanish derivations dataset we assign a new category according to the affix. We change all the affixes that end in *-ero*, *-ez*, *-ismo*,*-í* and *-illa* to N. We also assign V to those that contain the affixes *-ar* and *-ear*

```
affixes = ["-ero", "-ez", "-ismo", "-í", "-illa"]
condition = (
  df['category'].str.endswith(':U') &
  df['affix'].isin(affixes)
)
df.loc[condition, 'category'] = df.loc[condition, 'category'].str.replace(':U', ':N', regex=Fal

verb_endings = ["-ear", "-ar"]
condition = (
  df['category'].str.endswith(':U') &
  df['affix'].isin(verb_endings)
)
df.loc[condition, 'category'] = df.loc[condition, 'category'].str.replace(':U', ':V', regex=Fal

df[df["category"].str.endswith(":U")]
```

```
         pivot derivation category   affix
1281   demasía  demasiado      N:U    -ado
1595     mucho  muchísimo      U:U  -ísimo
1885     rueda     rodaja      N:U    -aja
4074       -és       -esa    ADJ:U      -a
9458     andar     ándale      V:U     -le
13821  demasía    demasié      N:U      -é
```

As a result we obtain 6 rows that can be eliminated from the final dataset because of all the mistakes they contain.

The second group of Spanish derivations (:U) cannot be easily fixed with a Python script, it contains many numerals and words that are not N, ADJ, ADV or V. Those rows that do not contain any of such categories can be dropped and the rest probably needs to be fixed manually. It contais some verbs, nouns and adjectives labeled with U, for instance *cuarenta cuanrentóń U:N -ón*. For some reason *cuarenta* is labeled in other rows as N but not in this one. Since numerals can be N or ADJ, alongside all the other issues with this group I think we can just drop all these rows (35). It is a low number that will not affect the results.

Polish data

Polish data seems to need more work as there are more incorrect labels, but the positive thing is that it can be fixed more easily. Affixes such as *-any*, *-ony*, *-ty*, *-y*, or *-ący*, *-ęty* take the label ADJ, because they are all endings that participles take.

```
affixes = ["-any", "-ony", "-ty", "-y", "-ący", "-ęty"]
condition = (
  df['category'].str.endswith(':U') &
  df['affix'].isin(affixes)
)
df.loc[condition, 'category'] = df.loc[condition, 'category'].str.replace(':U', ':ADJ', regex=
```

We can also see some formatting issues. Some rows under the same condition (X:U) contain the pivot and the derived form joined together in the pivot cell (i.e. *mylićpomylić pomylić*).

This can be fixed as well just removing the form from the pivot column and assigning to row the correct categories.

```
condition = df['category'].str.endswith(':U')
df.loc[condition, "pivot"] = df.apply(lambda row: row['pivot'].replace(row['derivation'], ''),
```

We also fixed certain rows that contained verbs in both columns but were not correctly labeled. This is fairly easy since in Polish verbs in the infinitive form end in *-ć* (most of them) or *-c*. We also labeled three rows incorrectly labeled U:U as ADJ:ADJ since they contained adjectives.

```
condition = (
  df["category"].str.endswith(":U") &
  df['pivot'].str.endswith("ć") &
  df['derivation'].str.endswith('ć')
)
df.loc[condition, 'category'] = "V:V"

condition = (
  df["category"].str.endswith(":U") &
  df['pivot'].str.endswith("c") &
  df['derivation'].str.endswith('c')
)
df.loc[condition, 'category'] = 'V:V'

pivots_to_change = ['zamężny', 'przystawalny', 'pocieszony']
condition = (
    df["pivot"].isin(pivots_to_change) &
    df["category"].str.endswith(":U")
)
df.loc[condition, 'category'] = 'ADJ:ADJ'

df_u = df[df["category"].str.endswith(":U")]
df_u.sample(15)
```

	pivot	derivation	category	affix
1875	co	cokolwiek	N:U	-kolwiek
1237	kształt	-kształtny	N:U	-ny
13250	czyż	czyżby	N:U	-by

2410	jaki	jakikolwiek	N:U	-kolwiek
12163	nieomal	nieomalże	U:U	-że
2378	dokąd	dokądś	U:U	-ś
13311	który	któryś	ADJ:U	-ś
2290	grecki	nowogrecki	ADJ:U	nowo-
14002	niejaki	niejako	ADJ:U	-o
378	jaki	jakiś	N:U	-ś
9709	przez	poprzez	U:U	po-
12103	coś	cośkolwiek	N:U	-kolwiek
10431	tak	takowy	ADV:U	-owy
23328	d	dż	N:U	-ż
15734	ile	ileś	U:U	-ś

The table above represents the rest, which can be removed as well as they do not contain any nouns, adjectives, adverbs or verbs. Take for instance the appearance of *co*, *kto*, *jaki*.. which are relative pronouns. Everything ending in *-ś* and *-ż* or *-że* are not nouns nor adjectives nor adverbs nor verbs, but other types of pronouns or particles, so they can be removed.

Regarding the data labeled as U:X, we can do much better than in Spanish because there are many rows (163) that contain verbs ending in *-ć* in both the pivot and the derivation column but are incorrectly labeled as U:V, for example *kręcić skręcic U:V s-* or *paść przepaść U:V prze-*. We can easily change the label to V:V.

```
condition = (
    df["category"].str.startswith("U:") &
    df['pivot'].str.endswith("ć") &
    df['derivation'].str.endswith('ć')
)

df.loc[condition, 'category'] = 'V:V'
df[condition]
```

	pivot	derivation	category	affix
91	kręcić	skręcić	V:V	s-
508	ąć	wziąć	V:V	wz-
555	padać	spadać	V:V	s-

```
1025     paść   przepaść      V:V  prze-
1502   stawać   dostawać      V:V    do-
...       ...        ...      ...    ...
18821   trzeć  rozetrzeć      V:V  roze-
18953    słać     dosłać      V:V    do-
18955    słać     obsłać      V:V    ob-
18962    słać    rozsłać      V:V   roz-
18993     jąć      dojąć      V:V    do-

[163 rows x 4 columns]
```

There are also 27 rows that contain verbs ending in *-c* in both the pivot and the derivation cells, which can be fixed just like previously done on the other group of verbs.

```
condition = (
    df["category"].str.startswith("U:") &
    df['pivot'].str.endswith("c") &
    df['derivation'].str.endswith('c')
)

df.loc[condition, 'category'] = 'V:V'
```

Finally we can fix some pivots that are verbs, but are not labeled as such, just by looking at the ending, although this needs to be done carefully as some nouns can also end in *-ć* or *-c*, so we are only doing it on the mislabeled ones (the ones labelled as U), which are all verbs.

```
condition = (
    df['category'].str.startswith('U:') &
    df['pivot'].str.endswith('ć') |
    df['category'].str.startswith('U:') &
    df['pivot'].str.endswith('c')
)
df.loc[condition, 'category'] = df.loc[condition, 'category'].str.replace('U:', 'V:', regex=Fa
df.loc[condition]
```

```
          pivot derivation category affix
```

```
223       ostać    ostatni    V:ADJ   -ni
3098   wiedzieć    wiadomy    V:ADJ  -omy
3121       rzec    rzekomy    V:ADJ  -omy
6497       paść     pastwa      V:N  -twa
12732     spaść   spasiony    V:ADJ  -ony
14938  wiedzieć     wywiad      V:N   wy-
```

Since we have fixed almost 200 rows, the resulting ones labeled as U:X contain only 21 rows, with some mistakes or words that are not N, ADJ, ADV or V so we can just drop them. Both resulting datasets do not contain any row labeled with U anymore.

Cleaning inflections

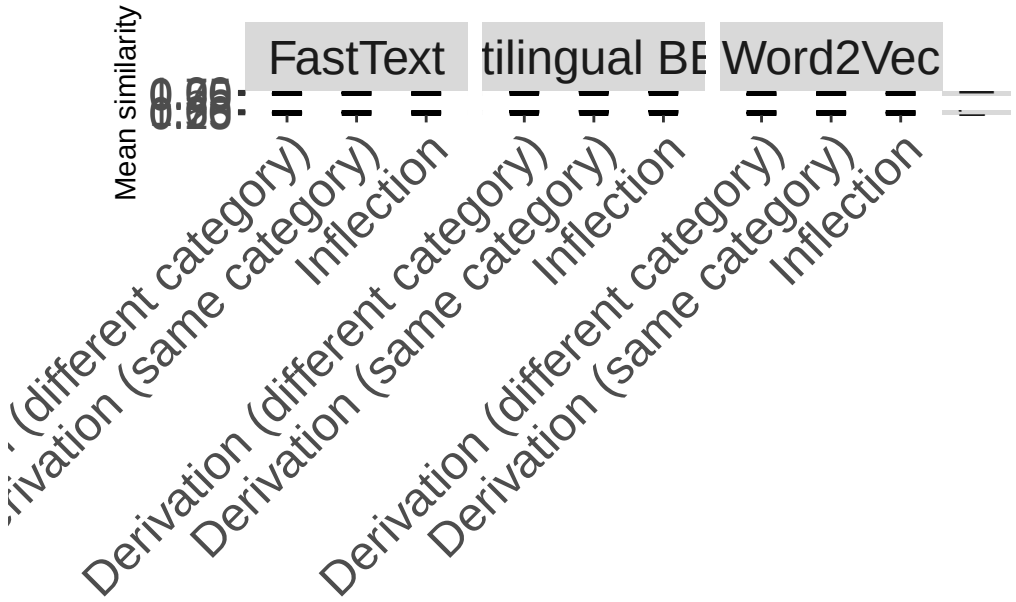
The only thing to clean in the inflections dataset are the *vos* and *usted* forms. This code was added to the filter_unimorph.py script.

```
df = pd.read_csv("C:/PythonCode/TFM_GLS/py/datasets/spa/spa.txt", sep="\t", header=None, names=
df = df[
    df["category"].str.contains("V;IND;PRS") | # presente
    df["category"].str.contains("V;IND;PST;IPFV") | # pret. impf.
    df["category"].str.contains("V;IND;FUT") # futuro simple
]

# removing vos forms
df = df[~((df['inflection'].str.endswith('ás') |
     df['inflection'].str.endswith('és') |
     df['inflection'].str.endswith('ís')) &
     df['category'].str.contains('V;IND;PRS'))]

# removing usted forms
df = df[~df['category'].str.contains('FORM')]
df.head(10)
```

```
   pivot inflection                 category
6   orar        oro            V;IND;PRS;1;SG
7   orar       oras       V;IND;PRS;2;SG;INFM
```

9	orar	ora	V;IND;PRS;3;SG
11	orar	oramos	V;IND;PRS;1;PL
12	orar	oráis	V;IND;PRS;2;PL
13	orar	oran	V;IND;PRS;3;PL
14	orar	oraba	V;IND;PST;IPFV;1;SG
15	orar	orabas	V;IND;PST;IPFV;2;SG;INFM
16	orar	oraba	V;IND;PST;IPFV;3;SG
18	orar	orábamos	V;IND;PST;IPFV;1;PL

Clean data results



The lowest category in both FastText and Word2Vec in Polish is V:ADV with a 0.08 and 0.09 mean similarity respectively.

Table 2: Mean similarity between pivot and inflection on the **clean data**

INFLECTION	WORD2VEC	FASTTEXT	MULTILINGUAL BERT
SPA	0.530683	0.550212	0.95

INFLECTION	WORD2VEC	FASTTEXT	MULTILINGUAL BERT
POL	0.513006	0.486620	0.91

Table 3: Mean similarity between pivot and derivation on the **clean data**

DERIVATION	WORD2VEC	FASTTEXT	MULTILINGUAL BERT
SPA	0.504823	0.511255	0.927637
POL	0.406783	0.544123	0.931497

The change in the means is minimal, less than 0.0001 but I guess we should get a cleaner mean by category in the plot.

Subset of the most frequent lemmas and affixes (after cleaning)

For this task in Spanish we used the 10000 most frequent lemmas in CREA. We extracted the verbs that appear in both datasets, UniMorph and CREA, and obtained a subset of 1568 lemmas from a total of 6695.

For the Polish data we used sgjp.pl. We filtered (using the site's implemented filter) the 8500 most common lexemes and took all the verbs from that list which were 1832. After that we compared that list of verbs to the UniMorph data and extracted those that appear in both datasets resulting in 455 lemmas from a total of 844 that appear in the UniMorph data.

To create the subset of affixes, we took the most common affixes in the UniMorph data itself.

UniMorph derivation data has 31252 rows in Spanish with 709 unique affixes and 58673 in Polish with 443 unique affixes. After creating a subset dataset of only the top 15 affixes in each language, the result is 14224 rows in Spanish and 33805 in Polish.

Table 4: Top 15 affixes in Spanish and Polish in UniMorph data.

(a) Spanish

Affix	Count
-mente	2997
-dor	1316
-ar	1310
-ero	1123
-miento	913
-ico	870
des-	836
-ción	831
-ear	676
-ista	642
-ito	638
-ismo	549
-ón	533
-idad	499
-al	491

(b) Polish

Affix	Count
-owy	5804
-ka	5487
-anie	3421
-ość	3287
-ny	2414
-ie	2161
-enie	1669
-ek	1521
-ować	1517
-o	1393
-ik	1249
-ski	1212
-ać	1158
-stwo	770
za-	742

Table 5: Mean similarity between pivot and form in inflection and derivation by model and language.

(a) Inflection

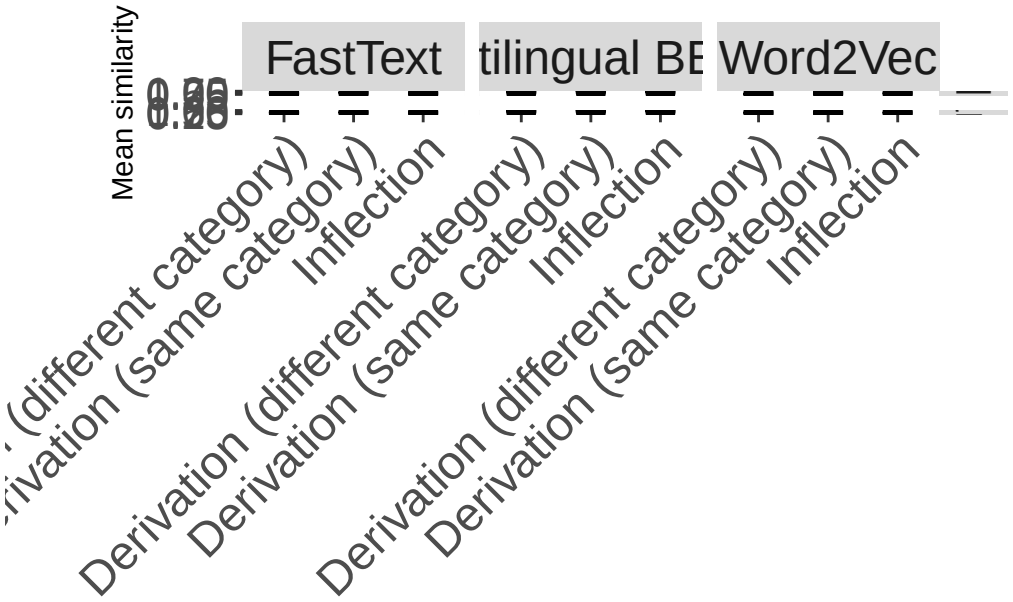
Model	Language	Initial Analysis	Clean Data	Subset Data
FastText	Spanish	0.550	0.511	0.523
	Polish	0.486	0.544	0.558
Word2Vec	Spanish	0.530	0.504	0.509
	Polish	0.513	0.406	0.401
Mult BERT	Spanish	0.950	0.927	0.931
	Polish	0.910	0.931	0.931

(b) Derivation

Model	Language	Initial Analysis	Clean Data	Subset Data
FastText	Spanish	0.497	0.511	0.523
	Polish	0.490	0.544	0.558
Word2Vec	Spanish	0.504	0.504	0.509
	Polish	0.435	0.406	0.401
Mult BERT	Spanish	0.927	0.927	0.931
	Polish	0.908	0.931	0.931

Subset data results

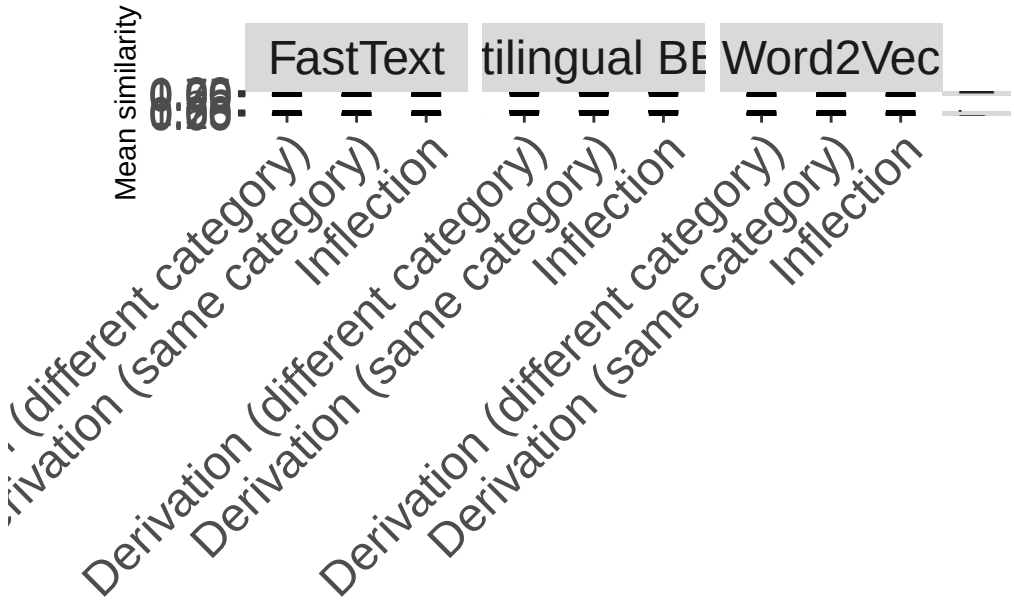
Here is presented a comparison of all the previous analysis and the one done on the subset data. Then a plot in **?@fig-subset** showing the mean similarity of each category on the subset data.



Random baseline

We made a random baseline in order to see wheter our methodology holds. For this task we randomly shuffled the data and ran the scripts once again. It consisted in maintaining the order of the pivots and shuffling the rest of the data 10 times in order to obtain a random baseline.

```
# FASTTEXT
# INFLECTION
spa_ft_inf <- read.csv("C:/PythonCode/TFM_GLS/py/results/spa/spa_fasttext_inflection_shuffled_
pol_ft_inf <- read.csv("C:/PythonCode/TFM_GLS/py/results/pol/pol_fasttext_inflection_shuffled_
# DERIVATION
spa_ft_der <- read.csv("C:/PythonCode/TFM_GLS/py/results/spa/spa_fasttext_derivation_shuffled_
pol_ft_der <- read.csv("C:/PythonCode/TFM_GLS/py/results/pol/pol_fasttext_derivation_shuffled_

# WORD2VEC
# INFLECTION
spa_w2v_inf <- read.csv("C:/PythonCode/TFM_GLS/py/results/spa/spa_word2vec_inflection_shuffled
pol_w2v_inf <- read.csv("C:/PythonCode/TFM_GLS/py/results/pol/pol_word2vec_inflection_shuffled
# DERIVATION
spa_w2v_der <- read.csv("C:/PythonCode/TFM_GLS/py/results/spa/spa_word2vec_derivation_shuffled
pol_w2v_der <- read.csv("C:/PythonCode/TFM_GLS/py/results/pol/pol_word2vec_derivation_shuffled

# BERT
# INFLECTION
spa_bert_inf <- read.csv("C:/PythonCode/TFM_GLS/py/results/spa/spa_bert_inflection_shuffled_res
pol_bert_inf <- read.csv("C:/PythonCode/TFM_GLS/py/results/pol/pol_bert_inflection_shuffled_res
# DERIVATION
spa_bert_der <- read.csv("C:/PythonCode/TFM_GLS/py/results/spa/spa_bert_derivation_shuffled_res
pol_bert_der <- read.csv("C:/PythonCode/TFM_GLS/py/results/pol/pol_bert_derivation_shuffled_res
```



One thing is apparent right away, the Multilingual BERT results do not make any sense.