Budapest University of Technology and Economics

Faculty of Natural Sciences

Department of Algebra

The impact of inflection on word vectors

B.Sc. Thesis

Dániel Lévai

supervised by

András Kornai, D.Sc.

full professor

Department of Algebra

Budapest University of Technology and Economics

2018

Contents

1 Introduction

In everyday life, we often use the word 'close' or 'similar' to describe the meaning of a word in relation to another word. For example, if we cannot recall the word 'puppy', we could say 'it is a dog, but smaller, younger'. In another situation, if we cannot remember the word 'tree bark', we could say 'tree shell', everyone would understand, because the meaning of 'shell' and 'bark' are somehow close in our minds. Going further, analogy tasks encode a certain 'direction' of the words e.g. 'man is to woman as king is to _____', and everyone can guess that the _____ is the word 'queen'. To give an other example, consider the following sentence: 'butcher is to knife as hairdresser is to _____'. We can identify the 'profession :: tool' relation as a relative position of words, giving us a 'direction', but the tool in the first case is also a cutter tool, so we tend to fill the blank with 'scissors', as it fits in both 'direction' and 'proximity'. Both examples treat similarity as semantic similarity, but one could solve morphological analogy tasks too, like 'can is to could as shall is to should'.

For a long time, linguists and mathematicians have tried to devise a method to assign a vector to each word, to embed words into vector space to be able to describe analogy tasks as finding the closest vector to $\overrightarrow{\text{king}} - \overrightarrow{\text{man}} + \overrightarrow{\text{woman}}$. Critically, most of these efforts, starting with Katz and Fodor (1963), assumed a discrete vector space (over GF(2) or GF(3)) and the idea to create continuous vector representations started decades later (Schütze, 1995; Bengio et al., 2003; Collobert et al., 2011). In the recent years, there has been a huge improvement in natural language processing (NLP) tasks with the use of such word embeddings, such as negative-sampling skip-gram models like word2vec (Mikolov et al., 2013a; Goldberg and Levy, 2014).

Due to the agglutinative nature of the Hungarian language, we have a

magnitude more words than inflected languages like English or Romance languages. With over 15 case endings only for nouns, and the possibility of prefixing and suffixing, we have many morphological ways to express ourselves. One could then ask the question whether the inflected form of the words are 'close' to the non-inflected form of the word, or whether the 'distance' of the words grow bigger as the word is inflected more heavily. To provide an example, one could say that *fa* 'tree' is similar to *fától* 'from the tree', while *fáinkról* 'of our trees' is more distant, because the suffixes 'pull away' the word from its nominative form.

In this thesis, we will be analyzing the impact of the words by inflection. In section 2, we will present necessary definitions and theorems which will be used in the latter sections. In section 3, we will explain how the word vectors are created, and why Mikolov et al. (2013a) has a huge importance in NLP. In section 4, we will demonstrate a method to process the morphological analyses produced by existing tools to be used in the following sections. At the end of the section, the measurements treating vector length and log-frequency in Arora et al. (2015) are verified. In section 5, a general overview is presented of the clusters established by the vectors of words with the same morphological analyses, and a similarity measure is defined between clusters. In section 7, we will compare the clusters by the affixes, measuring the deviation caused by the affixes.

The main result of this thesis is the defined similarity measure, which is proven useful for comparing word clusters and verifying the coherence of existing clusters while offering a slightly different word clustering. Linguistically, according to the model used, the current grammatical categories of the words are well-founded, with only exception to one part of speech.

2 Related theory

Definition 2.1. n-sphere

We can define the points of the n-sphere as the boundary points of the r-radius $n+1$-ball.

$$S^n = \{\mathbf{x} \in \mathbb{R}^{n+1} : \|\mathbf{x}\|_2 = r\} \tag{2.1}$$

Definition 2.2. Cosine similarity

The cosine similarity shows the cosine of the angle of two vectors.

$$sim_{cos}(\mathbf{u}, \mathbf{v}) = \frac{\langle \mathbf{u}, \mathbf{v} \rangle}{\|\mathbf{u}\|_2 \|\mathbf{v}\|_2}, \text{ where } \mathbf{u}, \mathbf{v} \in \mathbb{R}^n \tag{2.2}$$

Definition 2.3. Cosine distance of two vectors

$$dist_{cos}(\mathbf{u}, \mathbf{v}) = 1 - \frac{\langle u, v \rangle}{\|\mathbf{u}\|_2 \|\mathbf{v}\|_2}, \text{ where } \mathbf{u}, \mathbf{v} \in \mathbb{R}^n \tag{2.3}$$

The idea for the method of generating uniformly distributed random points came from Marsaglia, 1972, where the problem was stated for the 2-sphere.

Theorem 1. *Let X be a vector from an n-dimensional standard Gaussian normal distribution $(0, I_n)$, where I_n denotes the identity matrix of n dimensions. Then $\frac{X}{\|X\|_2}$ is uniformly distributed on the n-1-sphere.*

Proof. To see the uniformity of the distribution, we need to prove that X is invariant to orthogonal transformations. For any orthogonal matrix Q, $QX \sim \mathcal{N}_n(0, I_n)$, hence the distribution is invariant under any rotation. Let $Y = \frac{X}{\|X\|_2}$, then $Y_Q = \frac{QX}{\|QX\|_2}$. Since X is invariant to rotation, so is Y, and since $\|Y\|_2 = 1$ almost surely, then it must be evenly distributed on the sphere. □

Definition 2.4. Softmax function

The softmax or exponential normalized function is a logistic function that

enables to interpret a series of values as a probabilistic variable. Let $\mathbf{x} \in \mathbb{R}^n$ be a sample, $x_i \in \mathbf{x}$ a numerical observation. We can define the σ softmax function:

$$\sigma(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}} \quad \text{for } i \in 1, \ldots, n \tag{2.4}$$

Definition 2.5. Entropy

Information entropy is the average amount of information conveyed by an event, when considering all possible outcomes. Let x_i be events of X random variable.

$$\mathrm{H}(X) = -\sum_{i=1}^{n} \mathrm{P}(x_i) \log_2 \mathrm{P}(x_i) \tag{2.5}$$

3 Creating vectors from words

> You shall know a word by the company it keeps.
>
> *John Rupert Firth*

3.1 Vector space models

The idea of transforming text into vectors dates back to 1975 (Salton, Wong, and Yang, 1975), when Salton decided to create a large occurrence dictionary from multiple documents, then characterizing each document by the occurrences of the words from the vocabulary. The 'statistical semantics hypothesis' assumes that the word frequencies describe the meaning of the document. One could then measure the similarity of these document vectors by using their euclidean, cosine, Manhattan, or Jacquard distance. A document about cats must contain the word 'cat' with greater frequency, another document about domestic animals contains the word 'cat' with high frequency, so the respective coordinates for the two document vectors will be similar. One can even organize these vectors into clusters, creating groups of documents covering specific topics. Having document vectors enables querying a document database, we can create queries for specific phrases, thus allowing indexing and searching through large libraries. The so-called 'bag-of-words' hypothesis states exactly this phenomenon, a document's word frequencies show the relevance of the document to a query of the words. We can also calculate the tf-idf product 'term frequency – inverse document frequency' product for each word. If a word appears frequently in one document, it has high 'term frequency', and if that word appears only in a few documents, it has high 'inverse document frequency', the tf-idf product is high, so that

word is important for the document in a document collection (Turney and Pantel, 2010).

Taking this one step further, for each word, we could count how many times the other words from the vocabulary appears in a nearby window (in a certain length to each direction, usually chosen from 2 to 5). That way, we create a context vector for every word in the vocabulary, and as the quote says, the context of a word describes the word itself. Furthermore, the 'distributional hypothesis' states that words in similar contexts have similar meanings (Deerwester, Dumais, and Harshman, 1990; Harris, 1954). Spotting this among the word vectors is quite easy – if two vectors are similar by some similarity measure, the corresponding words have similar meaning.

This method seems easy and simple, but neural networks soon proven a better and more efficient way of generating word vectors than linear counting. (Mikolov et al., 2013b)

3.2 Skip-gram with negative sampling

The idea of neural networks dates back to the 40s (McCulloch and Pitts, 1943), when McCulloch created a computational network. The main idea is that our brain is composed of neurons (nodes) and synapses (edges). We learn and memorize by creating and strengthening synapses, and an artificial neural network – by analogy – should learn by strengthening and weakening weights on the edges based on the sample it receives. Constructing and training a neural network is a difficult task, because we do not have a strong idea how to interpret the weights of the edges or the nodes themselves – a neural network is a black box, and we do not always know how the architecture of the network should look like, or how we should train a network.

The architecture in a skip-gram model consists only of a single hidden

layer and an output layer with a softmax classifier. The task of the model is, for every word in the vocabulary, to learn the probabilities of every other word being in the context of the vocabulary word. In the following paragraphs, we will be using the figures of Chris McCormick to illustrate the model.

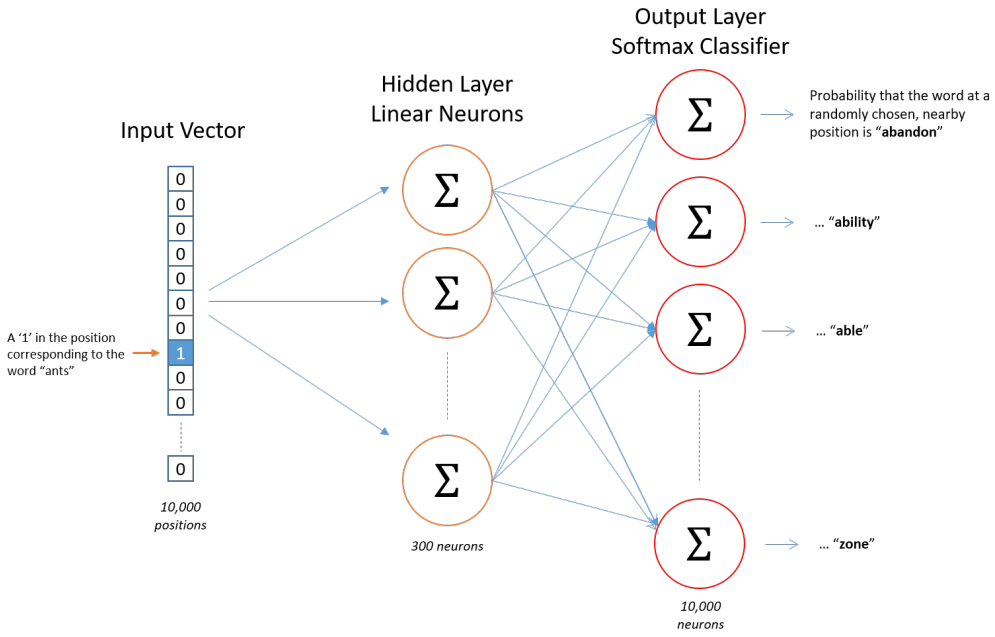

Figure 1: Network architecture

The input is a one-hot vector representing the word, and the output is a probability vector. As we can see, there are separate weights for each coordinate, and the number of nodes in the hidden layer defines the number of weights. If the hidden layer counts 300 neurons, then we will have 300 weights for each coordinate, thus for every word. To summarize, we create a model to do a fake task (predicting contexts) only to learn the input weights to be used as vectors. The same way, the model also learns the output weights and the classifation problem reduces to a matrix dot product and to a softmax classifation problem.

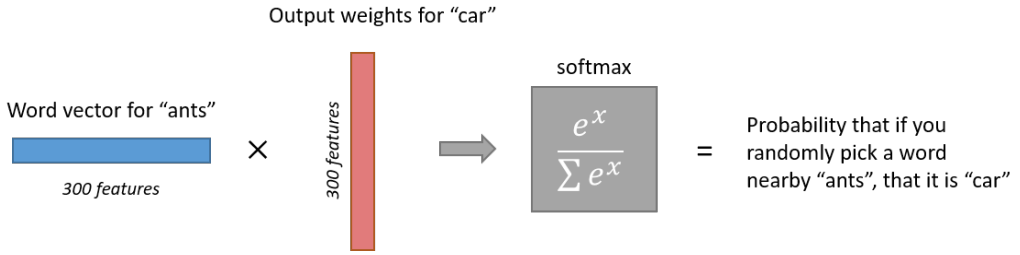

Figure 2: Classifying with skip-gram

For adjusting the weights, the model uses backpropagation, which is a method for refreshing the weights by calculating the partial differentials of the error caused by each weight. The main difference from the traditional neural networks are the subsampling, negative sampling, and the use of skip-grams, not continuous bag-of-words (CBOW). Skip-gram, in comparison to bag-of-words, predicts the context from the specific word (the model ‘skips’ the word), while bag-of-words predicts specific words from their contexts.

Subsampling is a technique to reduce the importance of the common words in the corpus. The intention behind the subsampling is that the words like ‘the’, ‘a’, ‘have’ occur very frequently, yet encode little semantic information about the context. The subsampling is probability based and uses the following function to determine the probability that the word w_i should be taken into account when updating the weights of the neural network, where z is the relative frequency function.

$$\mathbb{P}(w_i) = \left(\sqrt{\frac{z(w_i)}{0.00001}} + 1\right) \cdot \frac{0.00001}{z(w_i)} \tag{3.1}$$

Negative sampling is a technique to reduce calculation time. Without negative sampling, for each word, we would need to increase the weight of the edges of the correct guess (reward the specific edges for guessing the word right), and we would need to decrease the weights which predicted the context

wrong, so we would update every weight for every item in the vocabulary each training step. With negative sampling, we select a few noise words (5, in our case), and we update the network only by the error produced by these noise words. Thus for each training step, we would update the network 1+5 times.

3.3 gensim word2vec

The software we are using to create word embeddings is Radim Řehůřek's `gensim`. (Řehůřek and Sojka, 2010) It is a Python package created in 2010 to ease the usage of word embeddings, but ended up being one of the most robust, efficient and hassle-free softwares to realize unsupervised semantic modeling from plain text. While the software offers the possibility of fine-tuning every hyperparameter, by default it reduces noise, smooths vectors, and even removes words with low frequency and low semantic distinguishing value, so we used the default hyperparameters in the creation of our models: method of training is 'skip-gram with negative sampling', the dimension of the word embedding is 200, the window used is 5 words in both directions, and negative sampling is set to 5, meaning that for every training iteration, 5 'noise words' are drawn. The minimal samples were set to 5, meaning that the model automatically cropped rare words. The training consisted of 5 epochs, which was really fast, around 300000 words/second (while the corpus being gzipped) with Dual Xeon E5620 processors running at 2.40 GHz, it took around two and half hours for the whole corpus.

4 Manipulating the corpus

In the following section, we will offer a method to efficiently encode each word's morphological analysis without overly granulating the labels, while retaining the most of the underlying morphological structure.

4.1 Hungarian Webcorpus

The text corpus which we are using is the Hungarian Webcorpus, one of the largest Hungarian language corpora ever created, available in its entirety under Open Content license (Halácsy et al., 2004; Kornai et al., 2006). The corpus is based on 18.7 million pages crawled from the `.hu` domain. We are using the most noise-reduced subset, which has the duplicates, foreign language pages, and script-generated text (such as dates, headlines, tables of content) removed, leaving 710m word and punctuation tokens.

4.2 Morphological analysis and word normalization

In order to get the morphological analysis for each word, we used the e-Magyar digital language processing system (Váradi et al., 2017). E-magyar offers a toolchain for corpus lemmatizing, morphological analysis, speech processing, etc., however, in this thesis, we are only interested in obtaining every morphological analysis for each word, regardless of the context, and how easy it would be to disambiguate a word. For that, we used the emMorph module. (Novák, Siklósi, and Oravecz, 2016)

The emMorph module is relatively easy to use, nonetheless, the interpretation of the morphological analyses plays a vital role in understanding this thesis.

Word	Morphological analysis
számára	`szám[/N]a[Poss.3Sg]ra[Subl]`
számára	`szám[/N]ár[/N]a[Poss.3Sg][Nom]`
számára	`számára[/Post|(Poss)][Poss.3Sg]`
óriási	`óriás[/N]i[_Adjz:i/Adj][Nom]`
óriási	`óriási[/Adj][Nom]`
női	`nő[/N]i[_Adjz:i/Adj][Nom]`
női	`női[/Adj][Nom]`
női	`nő[/N]i[Pl.Poss.3Sg][Nom]`

We can see 3 fundamentally different analyses for the word *számára* here. The first line means that the word root is *szám*, a noun (`[/N]`), the following suffix is *a*, a third-person singular possessive suffix (`[Poss.3Sg]`), and the last suffix is *ra*, a sublative suffix (`[Subl]`), meaning 'onto his number'. The second line marks a compound word, more specifically, a noun-noun compound 'number-price' again in the possessive form 'his number-price'. The label marked by `/` codes the major category (part of speech) inflected with a third-person singular possessive suffix and the `[Nom]` marks the nominative case. The third line corresponds to the usual postpositional sense of the word, 'for him/her'.

Considering that we need to create a single token from multiple analyses, we need to devise a way of achieving that. We cannot simply concatenate these lines, because in the end, we want to have the same token for *női* 'women-related, female (adj.), his or her women' and for *óriási*. The solution is evident – we need to cut everything outside the square brackets, and then, concatenate them with a special marker (`<>`) indicating the ambiguous nature of the word. This still does not fulfill our expectations for a word with the label `[/N][Nom]<>[/Adj][Nom]` would be different than a word with the

`[/Adj][Nom]<>[/N][Nom]` label. To avoid this, we sorted the analyses in lexicographic order.

Compounds and derived words still pose problems. Consider the following words:

Word	**Morphological analysis**
elméleti	`elmélet[/N]i[_Adjz:i/Adj][Nom]`
elméleti	`elméleti[/Adj][Nom]`
együttműködés	`együtt[/Adv]működik[/V]és[_Ger/N][Nom]`
együttműködés	`együtt[/Prev]működik[/V]és[_Ger/N][Nom]`
együttműködés	`együttműködés[/N][Nom]`
számítógépes	`számít[/V]ó[_ImpfPtcp/Adj]` `gép[/N]es[_Adjz:s/Adj][Nom]`
számítógépes	`számít[/V]ó[_ImpfPtcp/Adj]` `gép[/N]es[_Nz:s/N][Nom]`
számítógépes	`számítógép[/N]es[_Adjz:s/Adj][Nom]`
számítógépes	`számítógép[/N]es[_Nz:s/N][Nom]`

Since it is unnecessary to store the fact that an adjective was derived from a noun, we can further simplify the variety of the labels. Furthermore, we can drop the analysis of the compound word until the last stem, meaning *számítógépes* 'computational, computer-related, person using a computer' becomes `[/Adj][Nom]<>[/N][Nom]`. *Együttműködés* 'cooperation' becomes `[/N][Nom]` and *elméleti* 'theoretical' becomes `[/Adj][Nom]`. To summarize the changes we have made so far:

Word	**Morphological analysis**
számára	[/N][Poss.3Sg][Nom]<>[/N][Poss.3Sg][Subl]<> [/Post|(Poss)][Poss.3Sg]
óriási	[/Adj][Nom]
női	[/Adj][Nom]<>[/N][Pl.Poss.3Sg][Nom]
elméleti	[/Adj][Nom]
együttműködés	[/N][Nom]
számítógépes	[/Adj][Nom]<>[/N][Nom]

Noise still remains in the corpus. We can notice a pattern concerning the numbers:

Word	**Morphological analysis**
60	[/Num|Digit][Nom]
13	[/Num|Digit][/Num|Digit][Nom]
1.5	[/Num|Digit][/Num|Digit][/Num|Digit][Nom]
10,5	[/Num|Digit][/Num|Digit][/Num|Digit][/Num|Digit][Nom]<> [/Num|Digit][/Num|Digit][/Num|Digit][Nom]
6-8	[/Num|Digit][Nom][Hyph:Hyph][/Num|Digit][Nom]
2002	[/Num|Digit][/Num|Digit][/Num|Digit][/Num|Digit][Nom]

In the sentence 'ten green bottles hanging on the wall', we can replace 'ten' with an arbitrary number without changing the apparent meaning of the sentence: 'there is *an arbitrary number of* bottles hanging on the wall'. We do not need to retain the fact that there is a dot, a comma, or a hyphen in a number. The rule we used to filter out these instances was the following: if [/Num|Digit][Nom] was the end of the morphological analysis, we have shrunk the analysis to 'digit, nominative case'.

Until this point, we have only written about the normalization of the morphological analyses, but seeing the next few examples, we need to construct

a way to eliminate the inconsistencies in our data.

Word	Morphological analysis	Freq
folytatni	[/V][Inf]	19456
folytatni.	[/V][Inf][Punct]	182
Isk	UNKNOWN	19
Isk.	[/N|Abbr][Nom]<>[/N|Abbr][Nom][Punct]	6047
feketére	[/Adj|col][Subl]	1100
Feketére	[/Adj|col][Subl]<>[/N][Subl]	28
nagy	[/Adj][Nom]<>[/Adv|AdjMod]	655076
Nagy	[/Adj][Nom]<>[/Adv|AdjMod]<>[/N][Nom]	107074
NAGY	[/Adj][Nom]<>[/Adv|AdjMod]<>[/N][Nom]	2531
nagy-	[/Adj][Nom][Hyph:Hyph]<> [/Adv|AdjMod][Hyph:Hyph]<> [/CmpdPfx][Hyph:Hyph]	1299
Nagy-	[/N][Nom][Hyph:Hyph]	257
nagy.	[/Adj][Nom][Punct]<>[/Adv|AdjMod][Punct]	134
-Nagy	[/N][Nom]	66
-nagy	[/Adj][Nom]<>[/Adv|AdjMod]	46
...nagy	UNKNOWN	40

As we can see, there are similar words with often similar morphological analyses, but with extra characters before or after them, and often with varying case. My algorithm to clear these inconsistencies is based on carefully grouping the lowercased form of the words by stripping the non-alphanumeric characters before and after the alphanumeric characters, and then merging words having the same stripped word form and the approximately same morphological analysis. The approximately same morphological analysis means that either the word has [Hyph:Hyph], [Punct]or [Hyph:Slash] in the end, which can be discarded, or the word's analysis is subset of a more frequent word's analysis with the same stripped form. In our case, *folytatni* 'to continue' will be merged with *folytatni.*, because they have the same stripped form and the latter has [Punct] in the end. The word *Isk* (abbreviation for

school) will be merged with *Isk.* because they share the same stripped form, the latter has higher frequency, and the former has `UNKNOWN` analysis. We cannot merge *Feketére* with *feketére*, because the capital cased variant has additional meaning ('onto the black one' vs 'onto a person named Black').

In the case of *nagy*, 'big, great', we have color coded the merges to make it clearer. First, they have the same stripped form not shown in the table, *nagy*, so we can concentrate on the analyses. Here, like in the previous case, the capital cased variant has the meaning 'Great', as shown in the morphological analysis `[/N][Nom]`. We can see the hyphenated word *nagy-*, which can be used in compounds: *nagy- és kisvárosok* 'big and small towns'. The boldface fonts mark that the word will remain in the corpus and that words having the same stripped form, similar analysis, and lower frequency will be merged into that word.

The effect of these simplifications is summarized in the next table:

	Before	**After**	**Reduction**
word types	7728127	7449116	3.61%
morphological analyses (labels)	139640	9245	93.4%
number of unknown tokens	6321835	5747920	9.08%

4.3 Obtaining word vectors

After obtaining the word embedding, we needed to measure the length of the vectors obtained. If we could project them to the surface of the unit sphere without great loss of information, we could compare them more easily.

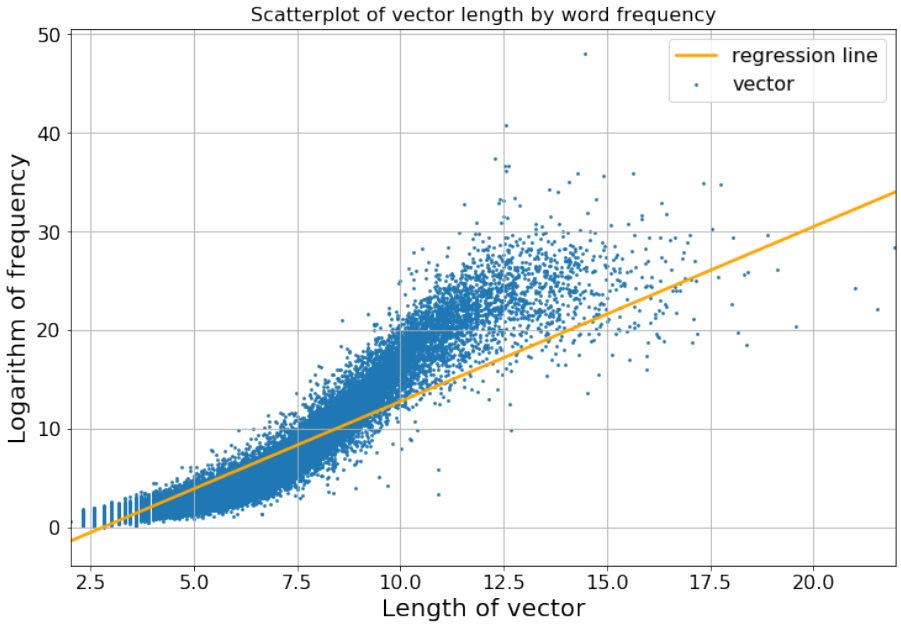

Figure 3: Length versus $\log_2$(frequency)

We can see a strong connection between the log-frequency of the words and the length of their vectors (Arora et al., 2015), so we have projected the words onto the surface of a 200-ball and added a set of 200000 uniformly distributed random vectors as a baseline for measurements.

An other characteristic of the skip-gram model is that it prefers placing the words in a specific part of n-dimensional space. It is hard to measure this because plotting the 2- or 3-dimensional PCA projection of the points lead to far not obvious figure. One technique to measure the spatial preference of the model is to count the relative frequency of each coordinate being positive, then plotting these numbers in an ascending order.

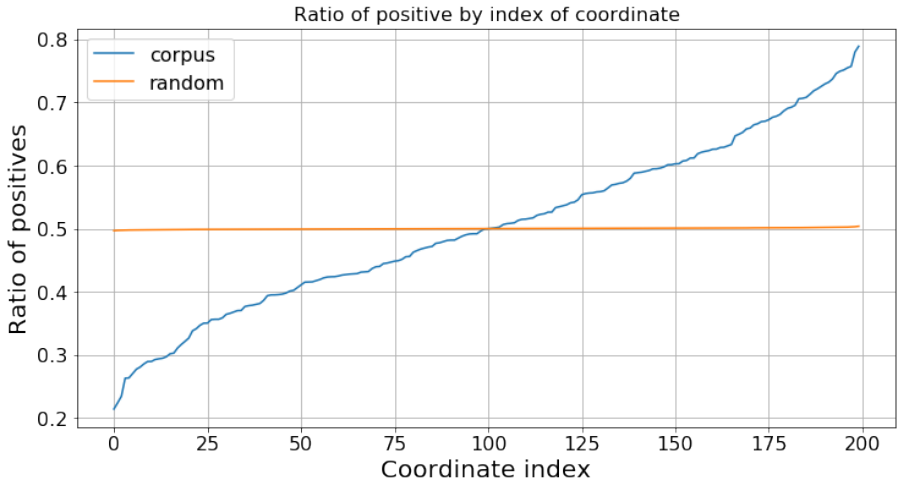

Figure 4: Probability of a certain coordinate being positive

The figure above shows that some coordinates are highly likely to be in the positive half of the space, while other coordinates are highly likely to be in the negative half of the space. If it were random, than the line would be flat - every coordinate would have 0.5 probability to be positive or negative.

5 Understanding word vectors

Justify my love.

Madonna

With our model trained, we could extract the hidden input weights from the neural network of the model – our data, the vectors. We suspect that there is a coherent structure in this vector space and each vector encodes a certain meaning and grammatical structure. We would like to justify that the clustering methodology we will be using in the following two sections is a viable approach to classify word vectors and that we can analyze these clusters in a way that helps understanding the word vectors.

We would like to emphasize the nature of this phenomenon, because the model used has no a priori knowledge of these grammatical categories, however, it will be seen that the clusters are indeed coherent in spite of the lack of grammatical knowledge.

5.1 Clusters

The problem a data scientist often faces is data presentation. In lower dimensions, we have techniques to visualize features, be it continuous, categorical, ordinal, with the addition of colors, using bigger or smaller dots in a scatter plot, using triangles for females, squares for males, we can visualize up to 4 or 5 dimensions without having an overly complicated figure. That is to say, if there is a slight hierarchy and we have some categorical features. In our case, having 200 equally important dimensions, we cannot select 3 and consider only those when we want to plot the data.

Nevertheless, we have tools from linear algebra to choose a mix of dimensions while retaining as much variance as possible from the original data,

and this method is called principal component analysis. Plotting a sample of 1000 vectors from the spherical projection of the first 3 principal components of certain cluster of word vectors yielded an interesting figure:

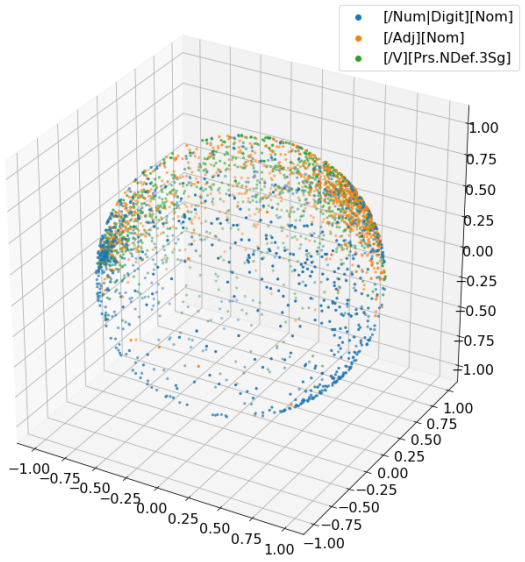

Figure 5: Clusters on the unit sphere

As we can see, each of these 3 clusters seem to have a core or a dominant quadrant – a dense area where most of the points are settled, and the points get sparser farther dense area. We need to verify if this phenomenon persists in the whole 200-dimensional space, and we need to find a way to identify a core and measure the density of these clusters.

5.2 Cluster statistics

To find out whether there is coherent structure of the word vectors, first, we need to analyze the clusters. One way of doing is the comparison of the standard deviations and the entropy of the clusters. If a cluster's standard deviation is high, it indicates low density, the lack of a core, and incoherent

structure. If the standard deviation is lower, it indicates a higher density, a more characteristic core. Plotting the clusters by their standard deviation on the x axis, and by the number of occurrences and their entropy on the y axis yields these two plots:

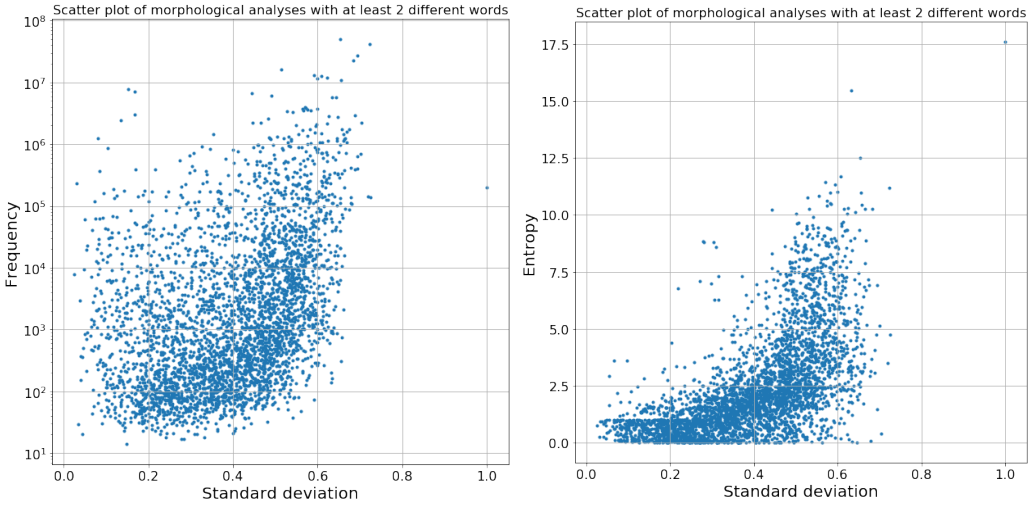

Figure 6: Scatter plots of clusters

On the first figure, we can see a square-like shape, showing weak correlation between the frequency and the standard deviation. The scatter plot of the entropy-standard deviation shows that higher entropy generally means higher standard deviation, however, after filtering out morphological analyses with low number of words, first 5, then 50, the plot showed a more circular, yet correlated shape:

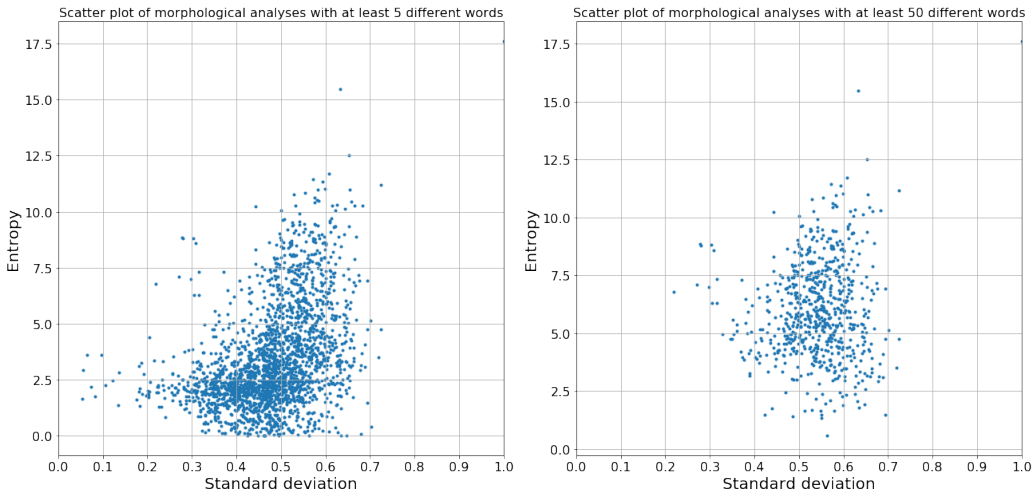

Figure 7: Scatter plots of clusters

We can spot a few outliers on this figure. The most notable in the upper-right corner is the `RANDOM` cluster, generated as stated in theorem 1, showing that our data is far less random than uniform random, the other two with high entropy are the cluster with the `UNKNOWN` label, and the cluster with 12.5 entropy is the `[/N][Nom]` cluster, mainly because it is the largest by frequency.

6 Quantifying similarity

In the following section, we are defining and giving intuition to a similarity measure between sets of vectors on the n-sphere, and applying it to the clustered word vectors.

6.1 Measuring the density

It is hard to have intuition in 200-dimensional space, especially on the surface of the 200-dimensional ball (called 199-sphere), but having the right intuition about the volume and the surface of the 200-ball is crucial in the evaluation of the following figures. First, let's see a theorem about the surface of the n-sphere, stated by Hopcroft and Kannan (2014):

Theorem 2. *For any $c > 0$, the fraction of the surface above the plane $x_1 = \frac{c}{\sqrt{n-2}}$ is less than or equal to $\frac{4}{c}e^{-\frac{c^2}{2}}$.*

Definition 6.1. n-cap
Let $\mathbf{m} \in S^n$, let $\alpha \in [-\pi, \pi]$. The cap defined by $\mathbf{m}$, α is

$$\text{cap}_\alpha(\mathbf{m}) = \{\mathbf{x} | \mathbf{x} \in S^n \wedge \langle \mathbf{m}, \mathbf{x} \rangle \geq \cos(\alpha)\} \tag{6.1}$$

which is equivalent to

$$\text{cap}_\alpha(\mathbf{m}) = \{\mathbf{x} | \mathbf{x} \in S^n \wedge \text{sim}_{cos}(\mathbf{m}, \mathbf{x}) \geq \cos(\alpha)\} \tag{6.2}$$

We can combine the theorem and the definition above to get an upper bound for the ratio of the surface of the cap and the n-sphere. Substituting $c = \cos(\alpha)\sqrt{n-2}$ into theorem 2, we get an upper bound of 0.4 for $\alpha = \frac{11\pi}{24}$ and 0.001447 for $\alpha = \frac{11\pi}{12}$. We can verify this upper bound by placing uniformly random points on the surface of the n-sphere and counting the points inside the cap, however, this method appeared to be unreliable for

20000 points on the surface of the 199-sphere, that is why the random sample was increased to 200000 points. With this method, we can finally measure and plot the ratio of points inside cap_α and compare this ratio to the random sample. To measure the compactness of clusters, we use an increasing cap around the cluster centroid, and plot the ratio of word vectors lying in the cap as a function of the minimal similarity of words to the cluster centroid. Note that the cap increases from the right of the figure to the left.

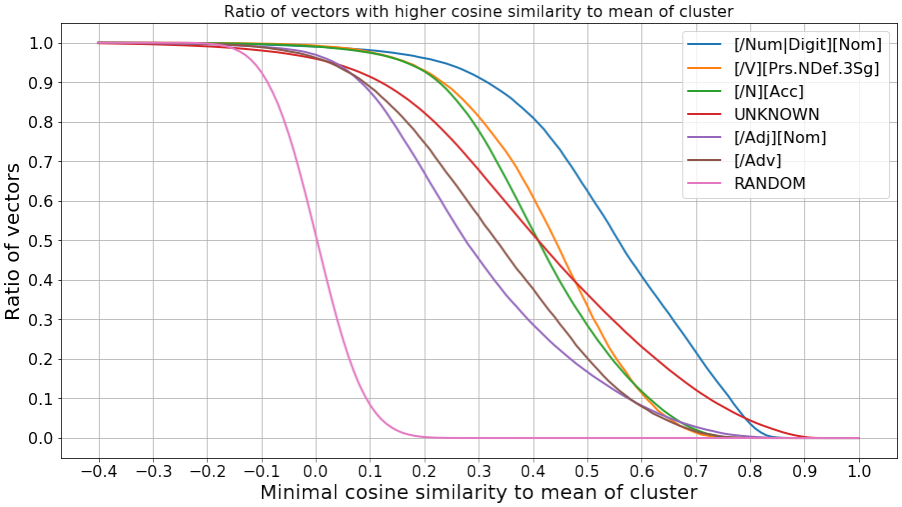

Figure 8: Ratio of points in a $\text{cap}_\alpha(\text{mean}_{\text{cluster}})$

As we can see in fig. 9, the `RANDOM` cap vanishes around $\cos(\alpha) = 0.2$ (for this α, theorem 2 limits the relative surface of the cap to 0.027), while the other clusters, most notably the `[/Num|Digit][Nom]` (digit in nominative case) shows the strongest coherence, which seems intuitive, as the numbers mostly indicate quantity, amount (counterexamples are dates, or symbolical numbers like 7, 3, 24/7). The `[UNKNOWN]` cluster shows high coherence, the reason can be the fact that this cluster is dominated by nouns. The `[/V][Prs.NDef.3Sg]` cluster (third person singular verbs) show the same coherence as the `[/N][Acc]` cluster (accusative nouns), while the `[/Adj][Nom]`

cluster (adjectives in noun case) show lower coherence than any of the clusters other than the `RANDOM` presented on the figure.

Since we want to filter out noise, and our ultimate goal is to measure similarity, we can use the ratio of the words in a cap_α with fixed α to measure self-similarity, and we can also calculate the ratio of some words in other clusters' cap. That way, we obtain a asymmetrical similarity measure. Obtaining the fixed α is based on filtering out the most noise, the most randomness. We use `RANDOM` as a base of comparison: in fig. 9, we show the ratios with that corresponding to `RANDOM` subtracted. Plotting showed us that the maximal difference is around $\cos(\alpha) = 0.13$, so we have chosen $\alpha = \frac{11\pi}{24}$ $(82.5°)$ to have a round number, because $\cos(\frac{11\pi}{24}) = \frac{1}{2\sqrt{2-\sqrt{2+\sqrt{3}}}} \approx 0.1305$.

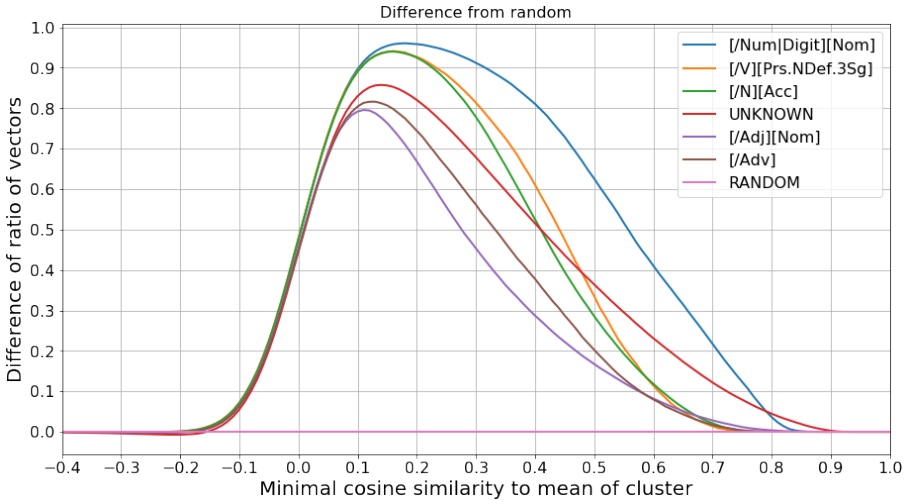

Figure 9: Difference of the ratios from `RANDOM`

6.2 Adjectives

In section 6.1, we have presented that the `[/Adj][Nom]` cluster has unexpectedly low coherence. Reviewing our methodology has shown us that the comparative and superlative forms of the adjectives were merged with the

non-comparative adjectives. This happened because we have cropped the derivation markers from the morphological analyses show in section 4.2. Consider the following examples:

Word	**Morphological analysis**
melegebb	`[/Adj][_Comp/Adj][Nom]`
legjobb	`[/Supl][/Adj][_Comp/Adj][Nom]`
legmegfelelőbb	`[/Supl][/Prev][/V][_ImpfPtcp/Adj][_Comp/Adj][Nom]`

The superlative affix is located at the beginning of the words in the Hungarian language. As a direct consequence, the affix is marked before the root, thus it was cropped. The comparative nature of the adjectives is also marked before the root as `[_Comp/Adj]`. The word *melegebb* 'warmer' poses no additional problem, we can use its analysis after cropping the `[/Adj]` from the beginning of the word. The superlative forms *legjobb* 'the best' and *legmegfelelőbb* 'the most suitable' are a bit more complicated. Due to the agglutinative nature of the language, a lot of affixes can appear between the `[/Supl]` (superlative affix) and the root. We cropped everything between them by the same reason we cropped in section 4.2, the fact that the word was derived is not important, we only need the comparative adjective nature of the word.

Replotting the figure presented in section 6.1, we can clearly see that the separate clusters of the comparative adjectives results in very coherent clusters, even better clusters, than the digits themselves. The `[/Adj]` cluster became only slightly more coherent, but this does not come as a surprise since only the 3.86% of the adjectives are comparative.

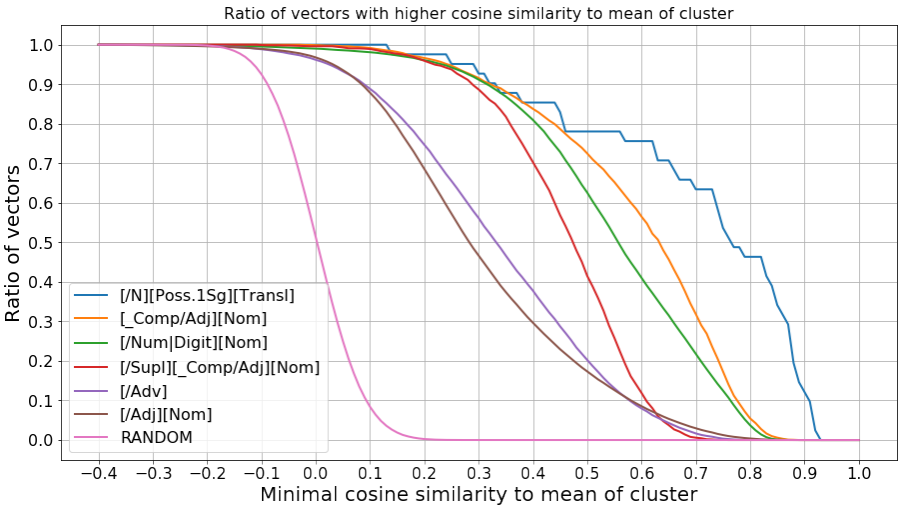

Figure 10: Ratio of points in a $\text{cap}_\alpha(\text{mean}_{\text{cluster}})$ with finer adjective clustering

7 The role of affix frequency

In the following section, we are examining the clusters based on their case endings to see whether some specific case endings contribute significantly more to the similarities than other case endings.

7.1 Self-similarities

We can use the defined similarity measure to see whether the clusters are justified. Consider the following clusters and their respective self-similarities:

Cluster	**Self-similarity**
`[/Adj][Nom]`	0.822
`[/Adj][EssFor:ként]`	0.904
`[/Adj][Supe]`	0.910
`[/Adj][Subl]`	0.924
`[/Adj][Acc]`	0.941
`[/Adj][Ade]`	0.945
`[/Adj][Ill]`	0.960
`[/Adj][All]`	0.978
`[/Adj][Transl]`	0.994
`[/Adj][EssFor:képpen]`	1.000
`[/Adj][Temp]`	1.000

We can see that the more specific case endings like `[/Adj][Transl]` and `[/Adj][Temp]` (translative and temporal case) show higher self-similarity, while the more general ones like `[/Adj][Nom]` and `[/Adj][Supe]` (nominative and superessive) show lower self-similarity. This tendency continues with the cases of noun, where `[/N][All]` and `[/N][Transl]` (allative and translative) are among the highest self-similarity cases and `[/N][Nom]` has one of the lowest self-similarity from the paradigm. More examples can be found on the next page.

Cluster	Sim	Cluster	Sim	Cluster	Sim
[/Adj][Nom]	0.822	[/N][EssFor:képp]	0.889	[/Num][Nom]	0.908
[/Adj][EssFor:ként]	0.904	[/N][Nom]	0.922	[/Num][Del]	0.955
[/Adj][Supe]	0.910	[/N][Ess]	0.926	[/Num][Dat]	0.957
[/Adj][Subl]	0.924	[/N][EssFor:ként]	0.936	[/Num][Ter]	0.960
[/Adj][Ine]	0.929	[/N][Ine]	0.937	[/Num][Cau]	0.971
[/Adj][Ela]	0.936	[/N][EssFor:képpen]	0.941	[/Num][Ill]	0.977
[/Adj][Acc]	0.941	[/N][Cau]	0.946	[/Num][All]	0.978
[/Adj][Ade]	0.945	[/N][Ade]	0.949	[/Num][Ine]	0.980
[/Adj][Ins]	0.951	[/N][Hyph:Hyph]	0.957	[/Num][Acc]	0.983
[/Adj][Abl]	0.959	[/N][Ter]	0.962	[/Num][Subl]	0.984
[/Adj][Ill]	0.960	[/N][Supe]	0.962	[/Num][Ela]	0.985
[/Adj][Cau]	0.961	[/N][Abl]	0.964	[/Num][Ade]	0.988
[/Adj][Del]	0.961	[/N][Acc]	0.966	[/Num][Ins]	0.992
[/Adj][Ter]	0.963	[/N][Temp]	0.966	[/Num][Abl]	1.000
[/Adj][Dat]	0.967	[/N][Ela]	0.968	[/Num][EssFor:ként]	1.000
[/Adj][All]	0.978	[/N][Del]	0.969	[/Num][Supe]	1.000
[/Adj][Transl]	0.994	[/N][Ill]	0.969	[/Num][Temp]	1.000
[/Adj][EssFor:képp]	1.000	[/N][Dat]	0.969	[/Num][Transl]	1.000
[/Adj][EssFor:képpen]	1.000	[/N][Subl]	0.969		
[/Adj][Hyph:Hyph]	1.000	[/N][Ins]	0.972		
[/Adj][Prs.NDef.3Sg]	1.000	[/N][Transl]	0.979		
[/Adj][Temp]	1.000	[/N][All]	0.979		
		[/N][Inl]	1.000		
		[/N][Prs.NDef.3Sg]	1.000		

7.2 Coherent clusters

We now return to clusters that are more frequent or have higher entropy (already shown in fig. 6). We partitioned the clusters into 20 equal bins by their respective standard deviation, then calculated the mean and the standard deviation (σ) of the vectors of each cluster. The clusters with difference from the mean by more than 2σ are the interesting clusters, meaning that their standard deviation is significantly lower than the typical cluster of the same frequency (fig. 12) or entropy (fig. 11). We can measure the difference from the mean in σ.

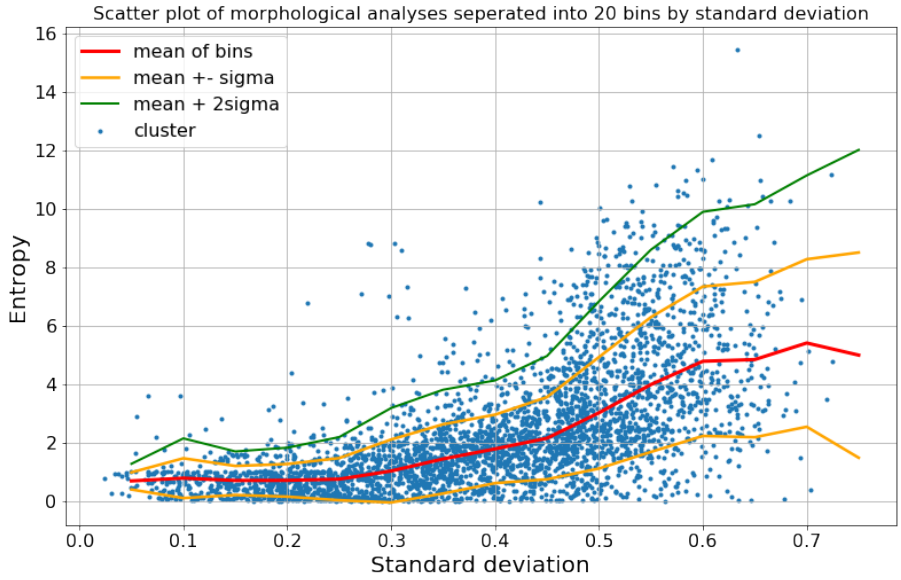

Figure 11: Binning clusters by standard deviation

On the figure, we can see that most of the points lay in the 1σ stripe, and the 2σ stripe is also rather populated. Each stripe is monotonically increasing. The interesting clusters are the ones above the 2σ stripe, because compared to their high entropy their variance is smaller than expected.

Taking a closer look at the bigger, non-ambiguous clusters (counting more

than 5000 words) outside the 2σ reveals that the nouns, be they plural or singular, form highly coherent clusters, and `UNKNOWN` shows the least coherence. The presence of infinitive and plural third person verbs among these most coherent clusters is very interesting, because verbs in general did not show strong coherence.

Bin	σ_{diff}	**Cluster**	$\textbf{Sim}_{\text{self}}$	**#Words**	**Frequency**
0.45	5.75	`[/V][Inf]`	0.988	14071	6677547
0.50	2.88	`[/Num|Digit][Nom]`	0.977	26666	6000563
0.55	2.25	`[/V][Prs.Def.3Pl]`	0.990	5230	1312497
0.55	2.56	`[/N][Pl][Subl]`	0.980	6795	582972
0.55	2.20	`[/N][Pl][Supe]`	0.979	5325	519220
0.55	2.72	`[/N][Pl][Ins]`	0.982	10135	955157
0.55	2.17	`[/V][Prs.NDef.3Pl]`	0.989	10486	3296388
0.55	2.94	`[/N][All]`	0.979	13858	1073443
0.60	2.60	`[/N][Ins]`	0.972	32886	3868455
0.60	2.37	`[/N][Del]`	0.969	12867	883380
0.60	2.44	`[/N][Pl][Nom]`	0.967	47068	11506915
0.60	2.28	`[/Adj][Pl][Nom]`	0.968	10934	1229296
0.60	2.57	`[/N][Dat]`	0.969	21785	1939670
0.60	2.42	`[/N][Subl]`	0.969	25687	3469518
0.60	2.21	`[/N][Pl][Acc]`	0.974	20702	3601070
0.60	2.25	`[/N][Abl]`	0.964	10270	649706
0.60	2.22	`[/N][Ela]`	0.968	12717	1028608
0.65	2.04	`[/N][Ade]`	0.949	7324	363706
0.65	3.99	`UNKNOWN`	0.892	199475	5643460
0.65	2.58	`[/N][Acc]`	0.966	61671	12617934
0.65	2.06	`[/N][Poss.3Sg][Acc]`	0.962	14164	2823258
0.70	2.48	`[/N][Nom]`	0.922	144945	50298170

We can look up the outliers by frequency the same way we did by entropy. The figure in this case looks a bit different, because we are not able to visualize frequency on a linear scale, thus the mean, which minimizes the

squared error of the points, is much higher on the figure than on the usual figures.

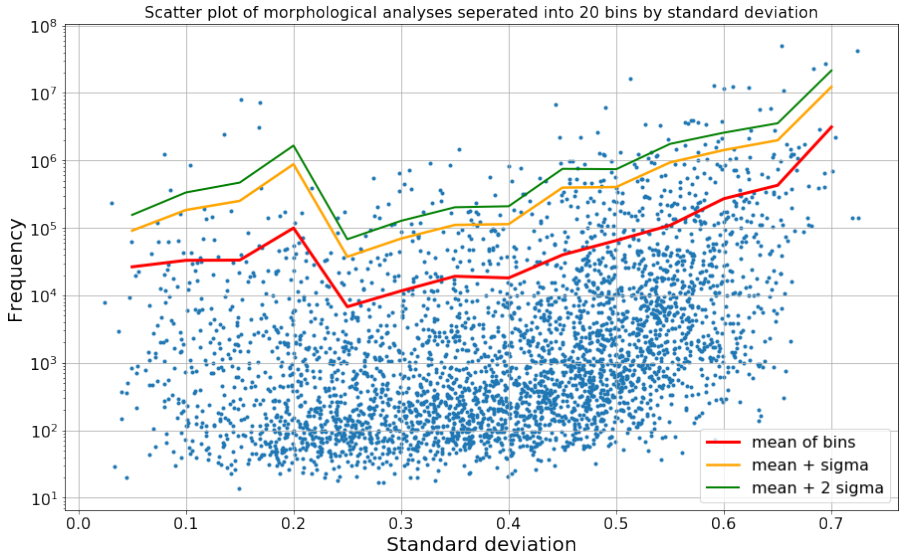

Figure 12: Binning clusters by standard deviation

There are less clusters outside the 2σ stripe, and most of them are also present on fig. 11, which can be expected since there is correlation between the entropy and the frequency. The singular, third person verbs appear in this list, and they also show high coherence. The entirety of the bigger, non-ambiguous clusters located outside the 2σ can be found below.

Bin	σ_{diff}	**Cluster**	$\textbf{Sim}_{\text{self}}$	**#Words**	**Frequency**
0.45	18.70	[/V] [Inf]	0.988	14071	6677547
0.50	17.57	[/Num|Digit] [Nom]	0.977	26666	6000563
0.55	3.87	[/V] [Prs.NDef.3Pl]	0.989	10486	3296388
0.60	3.12	[/N] [Ins]	0.972	32886	3868455
0.60	2.86	[/V] [Prs.Def.3Sg]	0.982	7369	3564999
0.60	9.76	[/N] [Pl] [Nom]	0.967	47068	11506915
0.60	2.78	[/N] [Subl]	0.969	25687	3469518
0.60	2.89	[/N] [Pl] [Acc]	0.974	20702	3601070
0.60	3.20	[/N] [Supe]	0.962	17881	3959914
0.60	11.11	[/V] [Prs.NDef.3Sg]	0.966	15483	13067569
0.65	3.44	[/N] [Ine]	0.937	24379	5801299
0.65	3.35	UNKNOWN	0.892	199475	5643460
0.65	7.81	[/N] [Acc]	0.966	61671	12617934
0.70	5.14	[/N] [Nom]	0.922	144945	50298170
0.70	2.66	[/Adv]	0.852	19591	27564415

7.3 Asymmetrical similarity

In section 6.1, we have already given the idea to compare one cluster's mean to another cluster's elements. When comparing not round-shaped clusters, this way of measuring similarity introduces asymmetry. Plotting a histogram of the differences (by subtracting the upper triangular matrix from the lower triangular matrix while preserving the direction of the comparison) shows a distribution quite close to normal distribution.

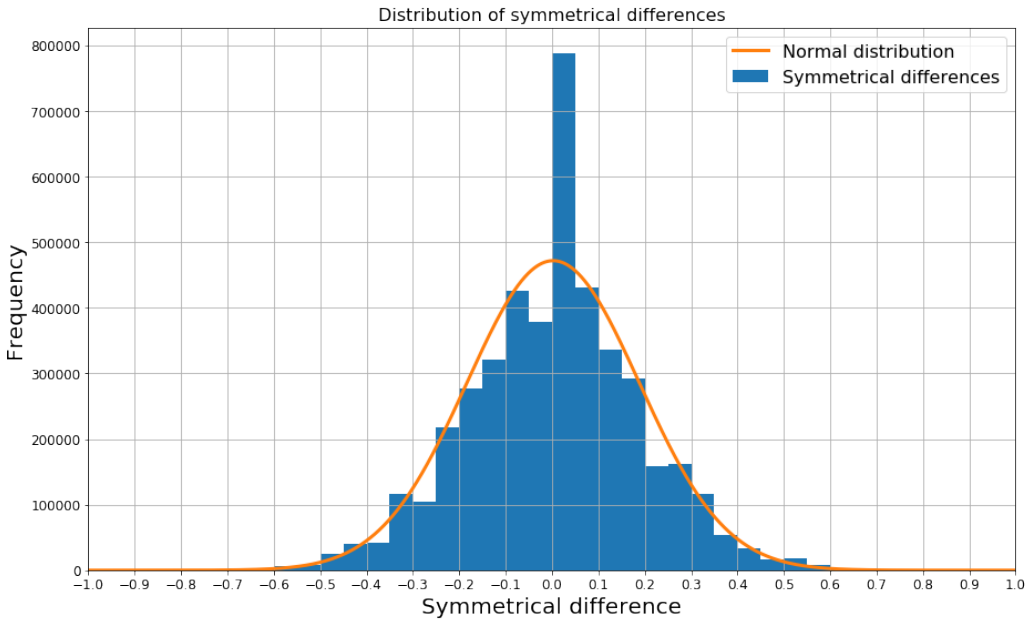

Figure 13: Distribution of symmetrical differences

Most of these differences are around 0, showing that most of the clusters are approximately round shaped. The two tails of the distribution are the important parts, because they show us pairs of clusters whose pairwise similarity in one direction is 1, while in the other direction this similarity is 0. One example to this phenomenon is the pair of `[/N|Pro][Subl][1Sg]` `[/N|Pro][3Pl][Dat]<>[/N|Pro][Poss.3Pl][Dat]` clusters. Both of the

clusters contain 4 vectors, but the words of the first cluster have 84456 occurrences in the corpus, and the words of the second cluster count 1090 occurrences. The words are pronouns in both cases, the first clusters' words are *énrám, reám, rám, énreám,* meaning 'onto me' with variable spelling, the difference is only stylistic, the words of the second cluster are *némelyiküknek, valamelyiküknek, mindegyiküknek, bármelyiküknek*, the first one meaning 'to some of them' and 'of some of them', while the rest meaning the same, but changing 'some' to 'specific one' , 'all', 'any'. The relation of 'valamelyiküknek' and 'bármelyiküknek' is strongly simplified in the previous sentence, but the explanation of this semantical difference is not the subject of this thesis. One reason for this strange phenomenon is that the *énrám, reám, rám, énreám* have identical meanings, the standard deviation of their cluster is very low, 0.04, while the other cluster of 4 words have significant difference in their meanings.

In the following sections, the asymmetry is of less importance. We can create a pairwise symmetrical similarity measure by taking the mean of their 2 similarities. As shown in fig. 13, most of the pairwise similarities have difference below 0.1, thus we do not lose much by symmetrizing the similarity measure.

7.4 Subcategories

E-magyar creates multiple subcategories for adjectives, nouns and numbers, and we can measure the pairwise similarity of their paradigms. If some subcategories show high similarity, we can say that it is not worth preserving as separate categories. Comparison of the subcategories to the `[/Adj]` categories yields interesting results.

Cluster$_1$	**Cluster$_2$**	**similarity**	**cases**
`[/Adj][.]`	`[/Adj][.]`	0.954	22
`[/Adj][.]`	`[/Adj|col][.]`	0.921	14
`[/Adj][.]`	`[/Adj|nat][.]`	0.900	16
`[/Adj][.]`	`[/Adj|Attr][.]`	0.865	7
`[/Adj][.]`	`[/Adj|Pro][.]`	0.843	18
`[/Adj][.]`	`[/Adj|Pro|Rel][.]`	0.549	7
`[/Adj][.]`	`[/Adj][Pl][.]`	0.884	17
`[/Adj][Pl][.]`	`[/Adj][Pl][.]`	0.956	17
`[/Adj][Pl][.]`	`[/Adj|col][Pl][.]`	0.949	9
`[/Adj][Pl][.]`	`[/Adj|nat][Pl][.]`	0.943	16
`[/Adj][Pl][.]`	`[/Adj][Poss.1Sg][.]`	0.855	10

`[.]` marks the pairwise comparison of single morphemes, so in the first few examples, we compare singular forms to singular form (because singular forms are not marked, thus a single morpheme after the word root must mean singular), and in the cases after, the plural forms. We can see a declining similarity when comparing more and more specific clusters, with the `[/Adj|col][.]` (adjectives describing colors) and `[/Adj|nat][.]` (adjectives describing nationality) being relatively similar to `[/Adj][.]`, while `[/Adj|Pro][.]` (pronominal adjectives) and especially the `[/Adj|Pro|Rel]` (relative pronouns like *amilyen* or *amekkora*, 'such as', 'as large as, as much as') show significantly less similarity. As indicated in section 7.1, more specific case endings may dominate the word vectors' similarity clusterwise, which is indeed the case in the last examples. Comparing plural adjectives, the similarities are significantly higher than their singular counterparts' similarities, while comparing singular to plural yields very low similarity.

7.5 Paradigm self-similarities

In the previous section, we have already used the `[.]` to indicate the comparison of paradigms. While the nominative forms may have lower similarities, the paradigm comparisons are dominated by the abundance of cases and case endings, producing very high self similarities. `[.]` denotes only a single morpheme, so this table aggregates only the 2-morpheme-long morphological analyses.

Cluster$_1$	**Sim**$_{\text{self}}$	**cases**
`[/Adj|Pro|Rel][.]`	1.000	7
`[/Num|Pro][.]`	1.000	9
`[/Num|Roman][.]`	1.000	6
`[/N|Acronx][.]`	1.000	13
`[/N|Pro|Rel][.]`	1.000	15
`[/Adj|col][.]`	1.000	14
`[/N|mat][.]`	0.998	17
`[/N|Ltr][.]`	0.997	13
`[/N|Abbr][.]`	0.996	13
`[/N|Pro][.]`	0.996	16
`[/Adj|Attr][.]`	0.995	7
`[/Adj|nat][.]`	0.995	16
`[/N|Unit][.]`	0.992	14
`[/V][.]`	0.989	54
`[/Post][.]`	0.987	8
`[/N|Abbr|ChemSym][.]`	0.986	6
`[/N|Unit|Abbr][.]`	0.984	14
`[/Num][.]`	0.979	18
`[/Num|Digit][.]`	0.975	14
`[/N|Acron][.]`	0.974	14
`[/N][.]`	0.958	24
`[/Adj|Pro][.]`	0.958	20
`[/Adj][.]`	0.955	22

8 Evaluation

Clustering word vectors by their morphological analysis has proven a good way to examine the impact of inflection on word vectors. We have tried statistical tests on the distances from the mean by cluster, and comparing them to the `RANDOM` cluster, but did not produce easily interpretable results, mainly caused by the high dimensionality. The 'cap similarity' on the other hand, while asymmetrical, has produced acceptable results, showed high coherence and similarity where expected, and showed lower similarity where difference was expected, thus justifying the selection of clusters for most cases. There are exceptions however, such as treating `[/Adj|Pro|Rel]` as a subcategory of `[/Adj]`, which our method shows to be mistake due to their low similarity.

9 Further research

Rothe, Ebert, and Schütze (2016) succeeded in creating meaningful ultra-dense subspaces for polarity, concreteness, frequency and part-of-speech (POS), supporting operations like 'give me a neutral word for *greasy*'. We could analyze the POS subspace, comparing the similarities of the clusters projected onto the subspace with the similarities obtained without projection. We are expecting interesting results from this projection.

Another idea of an exciting measurement arose at fig. 4. There is a stripe of coordinates with about 0.5 probability of being positive, we could discard these, about 50 or even 100 coordinates, reducing the dimension, and compare them to a model trained with the same number of dimensions. We are hoping to receive more distinct vectors thus more distinct clusters after this operation.

Other advancements are still to be elaborated, however, these two ideas already suggest a path for further studies.

References

Arora, Sanjeev et al. (2015). "Random Walks on Context Spaces: Towards an Explanation of the Mysteries of Semantic Word Embeddings". In: *arXiv:1502.03520v1* 4, pp. 385–399.

Bengio, Yoshua et al. (2003). "A Neural Probabilistic Language Model". In: *Journal of Machine Learning Research* 3, pp. 1137–1155. URL: http://www.jmlr.org/papers/v3/bengio03a.html.

Collobert, R. et al. (2011). "Natural Language Processing (Almost) from Scratch". In: *Journal of Machine Learning Research (JMLR)*.

Deerwester, Scott C., Susan T Dumais, and Richard A. Harshman (1990). "Indexing by latent semantic analysis". In: *Journal of the American Society for Information Science* 41.6, pp. 391–407.

Goldberg, Yoav and Omer Levy (2014). "word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method". In: *arXiv preprint arXiv:1402.3722*.

Halácsy, Péter et al. (2004). "Creating open language resources for Hungarian". In: *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC 2004)*. ELRA, pp. 203–210.

Harris, Zellig S. (1954). "Distributional structure". In: *Word* 10.23, pp. 146–162.

Hopcroft, John and Ravindran Kannan (Aug. 2014). *Foundations of Data Science*.

Katz, J. and Jerry A. Fodor (1963). "The structure of a semantic theory". In: *Language* 39, pp. 170–210.

Kornai, A. et al. (2006). "Web-based frequency dictionaries for medium density languages". In: *Proc. 2nd Web as Corpus Workshop (EACL 2006 WS01)*. Ed. by A. Kilgariff and M. Baroni, pp. 1–8.

Marsaglia, George (Apr. 1972). "Choosing a Point from the Surface of a Sphere". In: *Ann. Math. Statist.* 43.2, pp. 645–646. DOI: 10.1214/aoms/1177692644. URL: https://doi.org/10.1214/aoms/1177692644.

McCulloch, W.S. and W. Pitts (1943). "A logical calculus of the ideas immanent in nervous activity". In: *Bulletin of mathematical biophysics* 5, pp. 115–133.

Mikolov, Tomas et al. (2013a). "Distributed Representations of Words and Phrases and their Compositionality". In: *Advances in Neural Information Processing Systems 26*. Ed. by C.J.C. Burges et al. Curran Associates, Inc., pp. 3111–3119. URL: http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf.

Mikolov, Tomas et al. (Jan. 2013b). "Efficient Estimation of Word Representations in Vector Space". In: *Proceedings of Workshop at ICLR*. Ed. by Y. Bengio and Y. LeCun. Vol. 2013. arXiv: 1301.3781 [cs.CL].

Novák, Attila, Borbála Siklósi, and Csaba Oravecz (May 2016). "A New Integrated Open-source Morphological Analyzer for Hungarian". In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. Ed. by Nicoletta Calzolari et al. Portorož, Slovenia: European Language Resources Association (ELRA). ISBN: 978-2-9517408-9-1.

Řehůřek, Radim and Petr Sojka (May 2010). "Software Framework for Topic Modelling with Large Corpora". English. In: *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Valletta, Malta: ELRA, pp. 45–50. URL: http://is.muni.cz/publication/884893/en.

Rothe, Sascha, Sebastian Ebert, and Hinrich Schütze (June 2016). "Ultradense Word Embeddings by Orthogonal Transformation". In: *Proceedings*

of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. San Diego, California: Association for Computational Linguistics, pp. 767–777. arXiv: 1602.07572 [cs.CL]. URL: http://www.aclweb.org/anthology/N16-1091.

Salton, Gerard, Anita Wong, and Chung-Shu Yang (1975). "A vector space model for automatic indexing". In: *Communications of the ACM* 18.11, pp. 613–620.

Schütze, Hinrich (1995). "Distributional Part-of-Speech Tagging". In: *Proceedings of EACL*, pp. 141–148.

Turney, Peter D. and Patrick Pantel (2010). "From Frequency to Meaning: Vector Space Models of Semantics". In: *Journal of Artificial Intelligence Research* 37, pp. 141–188.

Váradi, Tamás et al. (2017). "`e-magyar`: digitális nyelvfeldolgozó rendszer". In: *XIII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY2017)*. Szeged.