UNIVERSITY OF TARTU
Institute of Computer Science
Computer Science Curriculum

Claudia Kittask

Computational Models of Concept Similarity for the Estonian Language

Bachelor's Thesis (9 ECTS)

Supervisor: Eduard Barbu, PhD

Tartu 2019

Computational Models of Concept Similarity for the Estonian Language

Abstract: The purpose of this thesis is to test and compare different computational models of similarity for the Estonian language. Models' predictions for words and concepts similarity is usually compared against human predictions. To make such comparisons between models' similarity estimates and human scores, a proper human annotated data set had to be created for the Estonian language. The SimLex-999 data set was chosen for translation into Estonian. This resource is used to test three families of computational models of similarity: distributional models, semantic networks and computer vision models. The results of this thesis can be used to evaluate future similarity models.

Keywords:
Semantic similarity, relatedness, computational models, distributional models, semantic networks, computer vision

CERCS: P170 Computer Science, numerical analysis, systems, control

Arvutuslikud mudelid eestikeelsete mõistetevaheliste sarnasuse leidmiseks

Lühikokkuvõte: Käesoleva bakalaureusetöö eesmärk on testida ja võrrelda erinevaid arvutuslikke mudeleid nende oskuse põhjal hinnata mõistete ja sõnade vahelist sarnasust. Mudelite hinnaguid võrreldakse inimeste hinnangutega. Selleks, et mudelite võimekust hinnata, luuakse uus eestikeelne andmekogu, mis sisaldab sõnapaare ja inimeste poolt annoteeritud sarnasuse hinnanguid. Töös hinnatakse kolme eri kategooriasse kuuluvaid arvutuslikke mudeleid: distributiivseid mudeleid, semantilisi võrke ja tehisnägemise mudeleid. Saadud tulemusi saab kasutada tulevaste mudelite hindamiseks.

Võtmesõnad:
Semantiline sarnasus, seotus, arvutuslikud mudelid, distributiivsed mudelid, semantilised võrgud, masinnägemine

CERCS: P170 Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine

Contents

1 Introduction

The semantic similarity between words is a useful property in many natural language processing tasks. Some examples of these tasks would be: word sense disambiguation, finding word spelling errors, machine translation, plagiarism detection and creating recommender systems.

There are many definitions for semantic similarity. Most commonly it is differentiated from the semantic association. Semantic association (also called relatedness) is a strength of the semantic interactions between two words. When determining relatedness there are no restrictions on the types of the semantic links used to determine this. Some authors define the semantic similarity as a subset of the notion of semantic relatedness where semantic links between two elements have to only be taxonomic [11]. This thesis is not using these definitions. Instead, the similarity between concepts is defined as a degree of synonymy, where synonymy is understood as a gradual relation between words.

Computational models of similarity for the English language have been evaluated and implemented in numerous studies. This is not the case for smaller languages such as Estonian. The reason behind this is that there are no human-annotated resources with similarity scores for Estonian. There are many human-annotated gold standard resources for English that are created to evaluate models of similarity, but these are not applicable for other languages. Many data sets created for evaluating models of similarity also have another problem: they do not differentiate similarity from the association. For example, WordSim-353 [6] rates words *coffee* and *cup* as similar to each other, even though these words do not share any common properties. This problem is resolved in the SimLex-999 [14] data set, which was chosen for translation into Estonian for the same reason.

There are some problems that are not fully addressed in previous works. First, many existing works do not differentiate between semantic similarity and semantic relatedness [35]. Second, typically only one type of computational model is used [35, 22] and no comparison of the different models are done. Third, the use of computer vision models to find semantic similarity is underexplored. Some studies [5, 4, 21] have used visual information with distributional semantic models, but using computer vision models alone is not well studied. In this thesis, these problems are addressed and discussed.

Now, when the human annotated resource for the Estonian language has been created, it is possible to evaluate different computational models for the Estonian language. Models that can differentiate similarity from relatedness can be used in different natural language processing tasks which models for relatedness cannot.

For example, machine translation and automatic creation of lexical resources are more suited for models of similarity.

This thesis has three goals. First goal is to create human annotated resource for Estonian language. This resource is useful because it enables to evaluate computational models performance, otherwise it would be impossible to know if the created model is any good. Second goal is to evaluate different computational models of similarity for the Estonian language on the newly created data set. Three categories of models are tested in this thesis: distributional models, semantic networks and computer vision models. It is important to evaluate different models to have a benchmark for future models. The third goal is to study how well computer vision models can estimate similarity.

The first chapter explains the concept of similarity and association and describes all the used computational models in this thesis. The second chapter describes the translation and re-scoring of the translated word pairs process. The third chapter contains information about the evaluation process of the models. In the fourth chapter, all the results are presented and discussed. The last chapter presents the conclusions.

2 Background

The aim of this chapter is to explain the meaning of semantic similarity and relatedness, discuss how computational models of similarity work and how these models can estimate similarity. Three categories of models are introduced in this chapter: distributional models, semantic networks and computer vision models.

2.1 Semantic Similarity and Relatedness

Semantic similarity is often confused with relatedness in the literature, but these terms are not identical. Semantic relatedness, which is also called association (Freud) in psychology, indicates the degree to which concepts are associated with each other. Concepts are highly associated if almost always these two concepts co-occur in space, time or language [31]. Semantic similarity is a special case of semantic relatedness and is at its strongest between synonym pairs.

These two terms can be best explained by an example concept pairs *plant-pot* and *plant-cactus*. Clearly, *plant* has nothing in common with *pot*, but it can be said that they are associated as they frequently occur together. *Plant* and *cactus* are semantically similar because they have common physical features (e.g roots, stems). In this case, *cactus* belongs to a category of plants.

Distinguishing these terms is important because models of similarity and models of relatedness have different applications in natural language processing. Models of similarity can be best used for tasks such as semantic parsing, machine translation and automatic creation of lexical resources. Models of relatedness are better for word sense disambiguation and text classification.

Additionally, it should be mentioned that there is difference between terms *word* and concept. The term *concept* refers to a specific sense of a given word. If two *words* are similar, this means that they denote similar *concepts*. For example, *right* and *correct* have the same meaning, but both express other concepts as well - *right* can also mean direction.

2.2 Distributional Models

Distributional semantic models use large text corpora to draw estimates of semantic similarities between words. These models are based on the distributional hypothesis [7], which states that words in similar contexts tend to have similar meanings as well.

To illustrate this hypothesis, it can be seen from the trivial example sentences *"Apples and pears are delicious fruits."* and *"There are a lot of apple and pear trees in the garden."*, that *apple* and *pears* often occur in the same context. Based on this information, these concepts can be perceived to be similar to each other.

2.2.1 Word2vec models

Mikolov et al. [25] developed the Word2vec method, which uses large data sets to learn continuous vector representations of words. These vectors can be obtained using two learning algorithms: continuous bag-of-words (CBOW) and skip-gram (SG). Both models are two-layer networks that train a classifier for a binary prediction task. The classification task is a bit different for SG and CBOW models. The predictions from the task are not used, instead, the models use the weights learned as an embedding.

The CBOW model uses the context of surrounding words to predict the word corresponding to this context.

CBOW architecture can be seen on the left side of Figure 2.1.

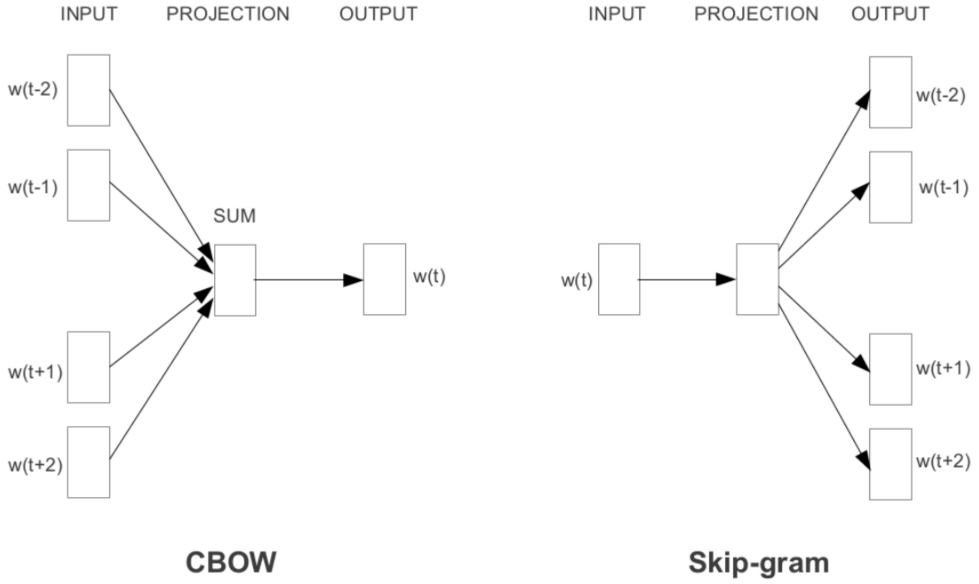


Figure 2.1. CBOW and Skip-gram architecture side by side. Image is from [25]

The skip-gram model is similar to CBOW, but instead of predicting the current word, the model predicts the context words from the current word. The Skip-gram model architecture is shown in Figure 2.1 on the right side.

There are some parameters that can be specified for these models. Parameters such as number of dimensions and context window size have been shown to greatly affect the model's ability to capture similarity between words.

Context window size is the number of words that are taken from both sides of the word to be included in the context. For example, figure 2.2 shows two different window sizes for a word *coffee*. According to Jurafsky[15], models with shorter window sizes typically can estimate similarity better than models with long window sizes. This comes from the fact that models with a shorter window size represent words more syntactically, as only the closest words are used. In comparison, longer window size models can estimate relatedness better.

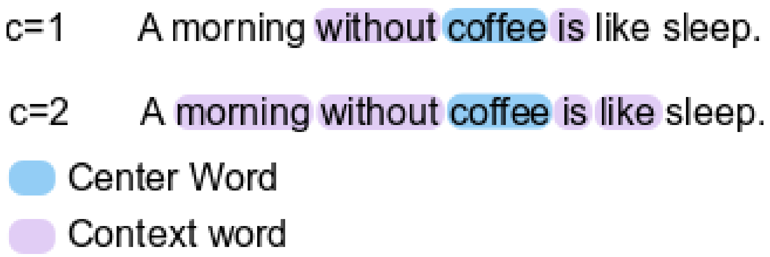

c=1
A morning without coffee is like sleep.
c=2
A morning without coffee is like sleep.
Center Word
Context word

Figure 2.2. Example of context window sizes

The number of dimensions is quite often chosen via trial and error or the default setting has been left unchanged. It was studied [29] how different dimension sizes can affect the model's performance. It was found that the model's performance was low until it reached to a certain number of dimensions (lower bound), after which it stabilized. The lower bound was computed from the number of pairwise equidistant words of the corpus vocabulary.

2.2.2 SenseGram

One problem with regular word embeddings is that they give the same vector representation for all senses of the word [13]. For example, the word *nail* can signify fingernail or also thin metal pieces. If one vector representation is used for such words, then likely only the prominent sense is reflected and other senses are neglected. This affects the similarity score between such words.

Pelevina et al. [30] introduced a method called SenseGram, that solves that problem by learning word sense vectors from pretrained word2vec word vectors. This method consists of four stages. First, the word embeddings are learned. Second, graph of nearest neighbours is built based on vector similarities. For that, graph of word similarities is created. For every word, 200 of its nearest neighbours is used. Third, now that every word is represented by a word cluster, this is used to construct an ego-network (2.3). This is done using graph clustering techniques. Clusters from that are interpreted as senses of the same word. Finally, sense embeddings are calculated for each sense in the induced inventory.

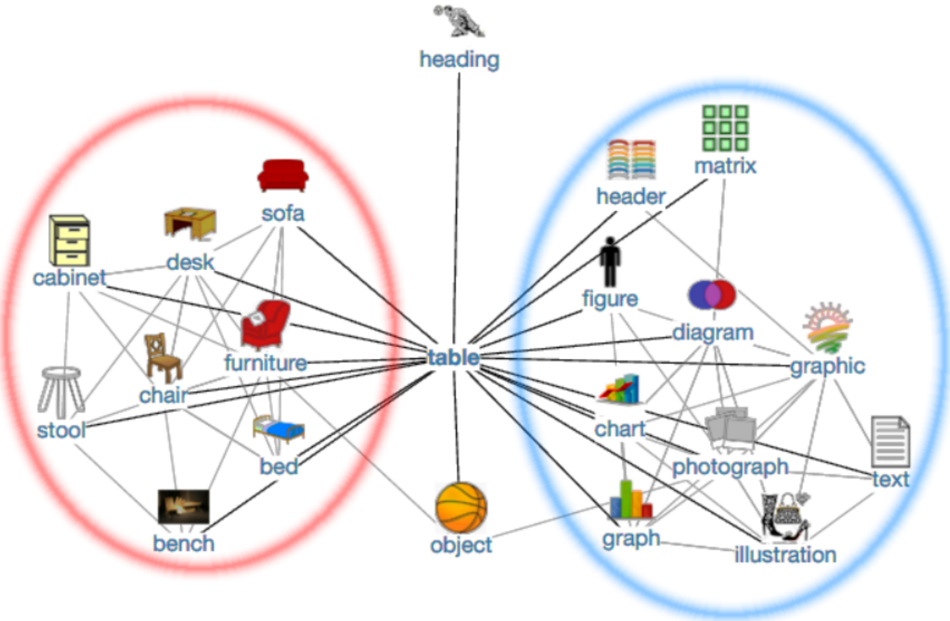


Figure 2.3. Visualization of the ego-network of *table*. Image is extracted from the article by Pelevina et al. [30]

Similarity between two words is calculated between all the possible senses and the maximum similarity score from all the calculated scores is returned by the model.

2.2.3 Similarity metric

To measure similarity between two vectors v and w, a similarity metric is needed. Usually, cosine similarity, which measures the angle between two vectors, is used. Next paragraphs are based on the discussion of vector similarity in Dan Jurafsky's book Speech and Language Processing [15].

Cosine similarity is based on the dot product operator:

$$dot - product(\vec{v}, \vec{w}) = \vec{v} * \vec{w} = \sum_{i=1}^{N} v_i w_i = v_1 w_1 + v_2 w_2 + ... + v_N w_N \qquad (1)$$

The dot-product is useful for the similarity measure because it is high when two vectors have large values in the same dimensions. It has one problem with longer vectors: if the vector is long, then the dot-product tends to be long as well. Vector length is defined as

$$|\vec{v}| = \sqrt{\sum_{i=1}^{N} v_i^2} \tag{2}$$

Frequent words have longer vectors, which means that more frequent words are more similar to each other. To overcome this problem, dot product is normalized, which is the same as the cosine of the angle between two vectors [15]. The cosine similarity between vectors $\vec{v}$ and $\vec{w}$ can be calculated as:

$$cosine(\vec{v}, \vec{w}) = \frac{\vec{v} * \vec{w}}{|\vec{v}|\,|\vec{w}|} = \frac{\sum_{i=1}^{N} v_i w_i}{\sqrt{\sum_{i=1}^{N} v_i^2}\sqrt{\sum_{i=1}^{N} w_i^2}} \tag{3}$$

The cosine value ranges from -1 to 1. Vectors that point in the same direction will have the cosine value 1, vectors that are orthogonal will have cosine 0 and vectors that point to the opposite direction will have the cosine value as -1. But because raw frequency values are positive then cosine for the vectors ranges from 0-1.

2.3 Semantic Networks

A semantic network represents knowledge as a graph. This graph contains nodes, which are representations of concepts (e.g ideas, events, situations, objects), which are connected through directed links. These links represent different semantic relations between the concepts [20].

Typically, only hypernym and hyponym relations (which are also called IS-A) are used for computing similarity between concepts. For example, *coffee* and *tea* in a semantic network would both belong to a broader category of *beverages.* This means that *beverage* is a hypernym of *coffee* and *tea.*

2.3.1 WordNet

WordNet [26] is a lexical inheritance database for English language created by the Cognitive Science Laboratory of Princeton University. It includes verbs, nouns, adjectives and adverbs and adds these to a separate set. Words in the WordNet are grouped to a sets of synonyms (synsets). A synset is a set, that contains synonymous words that express the same concept. For example, the entry for *coffee* includes synsets like:

1. S: (n) coffee, java
2. S: (n) coffee, coffee tree
3. S: (n) coffee bean, coffee berry, coffee
4. S: (n) chocolate, coffee, deep brown, umber, burnt umber

These synsets can be thought of as representations of a concept.

All synsets are linked with conceptual, semantic and lexical relations. The most important relations are hypernymy and hyponymy (IS-A) links. Each synset is related to it's more general and more specific synsets. This path to more general synsets can be followed all the way up to a root node. Figure 2.4 shows a fragment of noun IS-A relations in the WordNet.

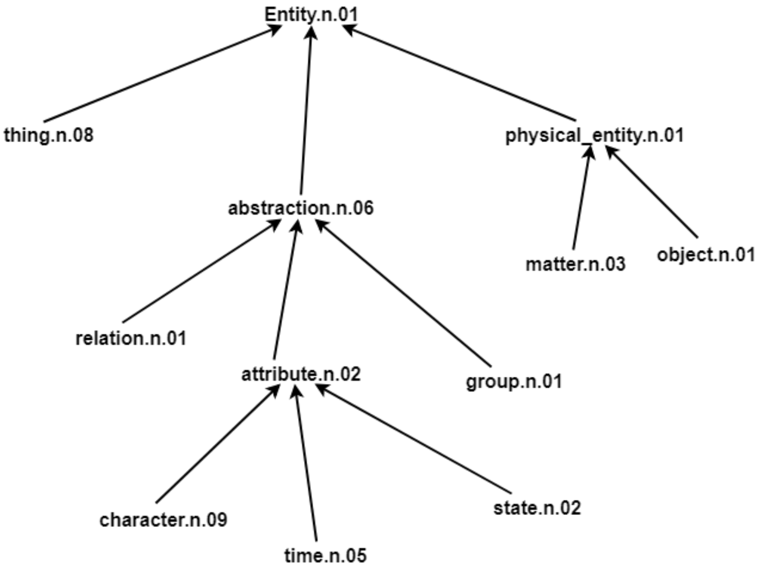


Figure 2.4. Part of WordNet IS-A relations

2.3.2 Wikipedia Page and Category taxonomy

Wikipedia is a human collaborative effort for producing a multilingual, free-content encyclopedia [24]. At the moment it covers 303 different languages [1]. It is an exceptional resource with its manually added concepts and relations.

[1] This information is taken from https://en.wikipedia.org/wiki/Wikipedia

Wikipedia can be used for automatically deriving hypernymy information for the concepts inside Wikipedia. Flati et al. [8] describe the process of automatic creation of integrated taxonomies for Wikipedia pages and categories. This method called MultiWiBi is applicable to different languages.

To understand this method they used, it is necessary to explain some terms. Wikipedia consists of pages and categories. A Wikipedia page gives an encyclopedic overview of a concept or an entity. This page also contains links to other pages, which makes these pages associated with each other. Wikipedia categories are entities that divide pages into broader classes. Usually, there are multiple categories for one page. These page-category associations are referred to as cross-links. Due to these links, the hypernymy information extracted from the page side can be transferred to the category side. There are pages with no assigned category and categories with no pages. Besides pages, there are also redirections, which are special pages acting as HTML redirections to other Wikipedia pages. For example, *Kaktus* redirects to the page *Kaktuselised* instead. Another important term to define is sense inventories, which are predefined sets of concepts. They form the sense inventory by using all the Wikipedia pages, categories and redirections. Hypernyms for pages are from the set of pages and redirections and hypernyms for categories are taken from the set of categories.

Now, it is possible to describe the process of creating Wikipedia bitaxonomy. First, the initial page taxonomy is made by parsing the textual definitions from the pages and extracting the hypernym lemmas, all these extracted lemmas are disambiguated using the sense inventory. For every page, the best generalization lemma is determined. Usually, this is extracted from the first sentence of the Wikipedia page, which usually provides a definition for the page. For example, the first sentence for Wikipedia's page *Tartu "Tartu is the second largest city of Estonia, after Estonia's political and financial capital Tallinn."* [2], tells that *Tartu* is a city. Second, the hypernyms in the page taxonomy and their links to categories are used to create a taxonomy for Wikipedia categories.

As mentioned earlier, MultiWiBi is applicable to other languages. Extracted taxonomies from different Wikipedias can also be browsed online [3]. Figure 2.5 is a screen-shot taken from the website showing a fracture of Estonian Wikipedia page and category taxonomy and links between them for *Tartu*.

[2] https://en.wikipedia.org/wiki/Tartu

[3] http://wibitaxonomy.org/

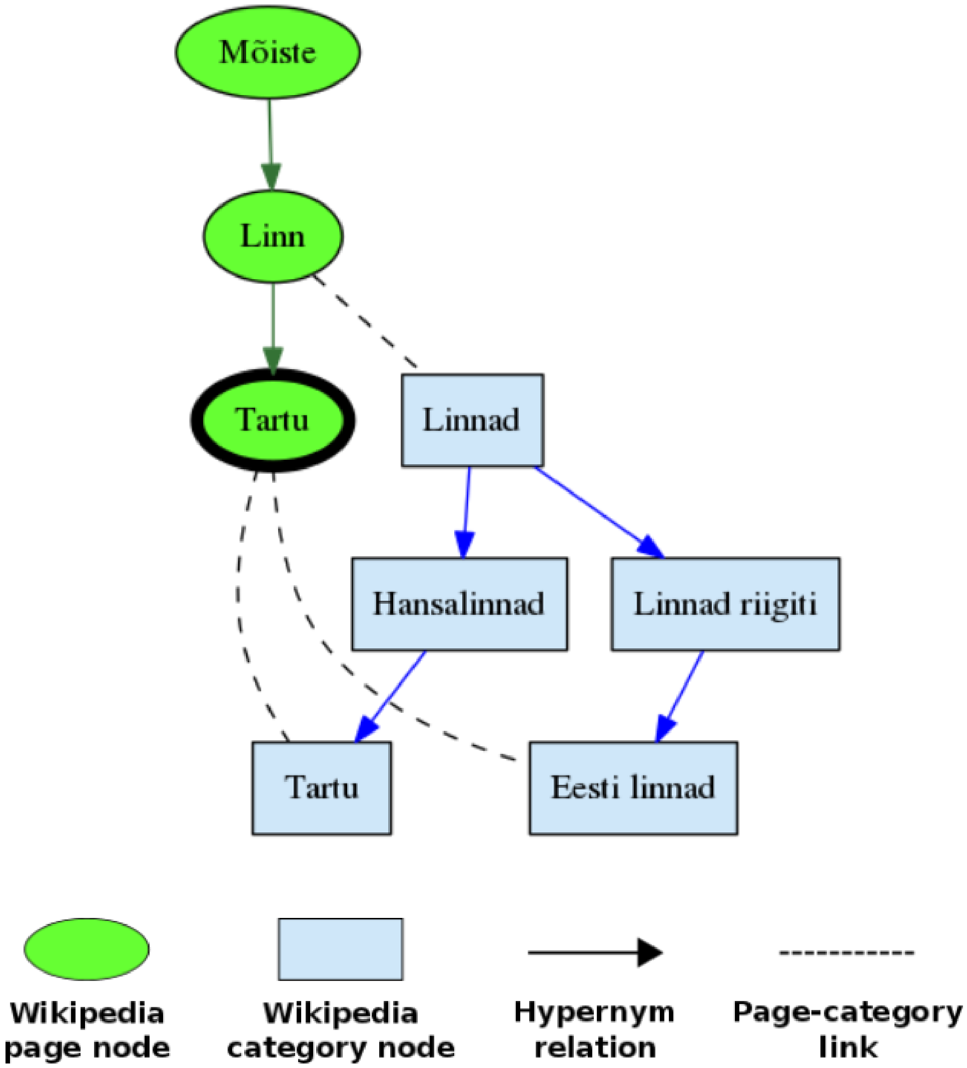


Figure 2.5. Screen-shot of online Wikipedia page taxonomy

2.3.3 Path-Based Similarity measures

In the semantic network framework, word similarity is derived from path length. Path-based measures find the shortest path between two nodes in a hierarchical semantic network. Intuitively, a shorter path between two nodes means higher similarity between them and vice versa.

Rada et al. (1989) [32] defined conceptual distance between two nodes as shortest path connecting those nodes in a taxonomic tree. This measure (Rad) counts the number of edges between two nodes. Edges have to represent classical lexical relations (e.g hypernyms, hyponyms). Rad can be computed as

$$dist_{Rad}(c_1, c_2) = l(c_1, c_2) \tag{4}$$

where *l(*c_1*,*c_2*)* returns the number of edges between c_1 and c_2.

Expressing similarity from the distance can be converted with a formula:

$$sim_{Rad}(c_1, c_2) = \frac{1}{dist_{Rad}(c_1, c_2) + 1} \tag{5}$$

This is often referred just as a path similarity measure in different studies.

Leacock and Chodorow (LC) [18] introduced non-uniform edge weighting measure, that uses logarithmic transformation to normalize the path length with the depth of the graph:

$$sim_{LC}(c_1, c_2) = -\log \frac{l(c_1, c_2)}{2 * depth} \tag{6}$$

where depth is the length of the longest path from the root node to a leaf node. The length $l(c_1, c_2)$ is measured in nodes [35].

There is a problem with these approaches: they are based on a notion that links represent even distances on the taxonomy [33]. Some links that are deeper inside the taxonomy tree often represent an intuitively narrow distance. Other links, which are closer to the root node, represent a wider distance [15].

This problem is taken into account in the Wu and Palmer (WuP) [34] similarity measure. This measure uses lowest common subsumer (LCS) of two concepts. LCS is defined as the first shared concept on the paths from the concepts to the root concept. WuP can be computed as

$$sim_{WuP}(c_1, c_2) = \frac{2 * depth(lcs)}{l(c_1, lcs) + l(c_2, lcs) + 2 * depth(lcs)} \tag{7}$$

2.4 Computer Vision models

Human semantic knowledge does not rely only on verbal and lexical information, but also on perceptual information. Learning image similarity is challenging because it has to capture between-class and also within-class image differences. For example, dog is similar to the cat because they have common features such as fur, eyes, four legs, similar shape, etc. Computer models use this visual information to compare images based on their similarity.

In this subchapter, two computer vision models are described. These models are convolutional neural networks and convolutional autoencoders.

2.4.1 Convolution Neural Network

These next paragraphs are mostly based on the book by Goodfellow et al. [9] if not stated otherwise. Convolutional neural networks (CNN) [19] are deep neural networks in which convolution is used at least in one of the network layers. CNNs use filters, also called kernels, to process data. For images, filters can be thought as a 2D grid of pixels that slide over the image. These filters are calculating dot-products, which are added to the feature maps. The resulting feature maps are used as an input for the next layers.

CNNs are particularly good at classifying images. The first convolution layer learns to detect simple features like edges and corners. Following layers learn more complex features by combining previous simpler ones [1]. Therefore CNNs can learn to recognize high-level image features, which can correspond to human language semantic description of the objects.

The idea behind using CNNs for semantic similarity computation is that when they are trained on large data sets, it is possible to extract the semantic representation of concepts from the deeper layers.

2.4.2 Convolutional Autoencoder

Convolutional autoencoders (CAE) are a hierarchical unsupervised feature learning extractor. The CAE model is based on autoencoder (AE) model, which is fully connected neural network that attempts to copy its input to its output. Autoencoders are often used in image compression, where first the image is encoded and later decoded. The AE network consists of three parts: an encoder, bottleneck (also called latent space or the hidden layer) and decoder layer that produces the reconstruction of the input [23].

AE is useful when it does not learn to copy the input exactly. To ensure this, the AE is constrained by a small bottleneck while training, this ensures that the model only learns the most useful properties of the training data [9].

Compared with AE, CAE model uses convolution and pooling layers in front and behind the fully connected layers. This facilitates the downsampling and upsampling of the data. Figure 2.6 shows the common CAE model architecture.

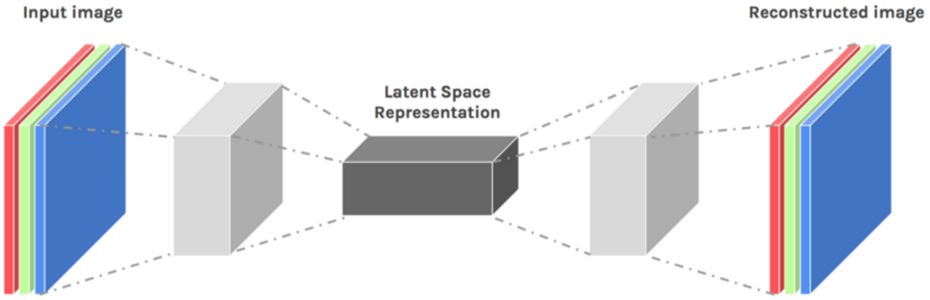


Figure 2.6. Sample architecture of an CAE. Image extracted from [17].

The main idea of using CAEs for similarity is that these sparse representations are close in the embedding space for similar concepts.

3 EstSimLex-999

To evaluate different computational models, a data set with human annotated scores is needed. There currently is no similarity set for the Estonian language. This chapter gives an overview of one of the best human annotated similarity sets in English, which was chosen for translation into Estonian. This chapter also describes the translation and re-rating process in detail. The translated data set can be accessed from the public GitHub repository. The link for it can be found from the Appendix.

3.1 SimLex-999

To evaluate models for semantic similarity it is necessary to have a data set, which contains similarity scores for word pairs. One way for getting these similarity scores is to let humans rate similarity between the word pairs. Many data sets have been created like this. The next paragraphs, which are based on the article by Hill et al. [14], introduce one of such data set called SimLex-999 .

The SimLex-999 data set gives values on a scale 0-10. This set contains 999 human annotated word pairs. Table 3.1 shows an example of similarity scores from SimLex-999.

This data set is considered hard for computational models to replicate because the model has to capture similarity independently of relatedness. SimLex-999 contains many pairs, such as *movie-theater*, that are strongly associated but not similar. This is hard because most corpora based representation-learning models learn connections between words from their co-occurrence in the corpora, which reflects relatedness more than similarity.

word 1	word 2	POS	Similarity
old	new	A	1.58
boy	kid	N	7.5
brother	soul	N	0.97
find	disappear	V	0.77

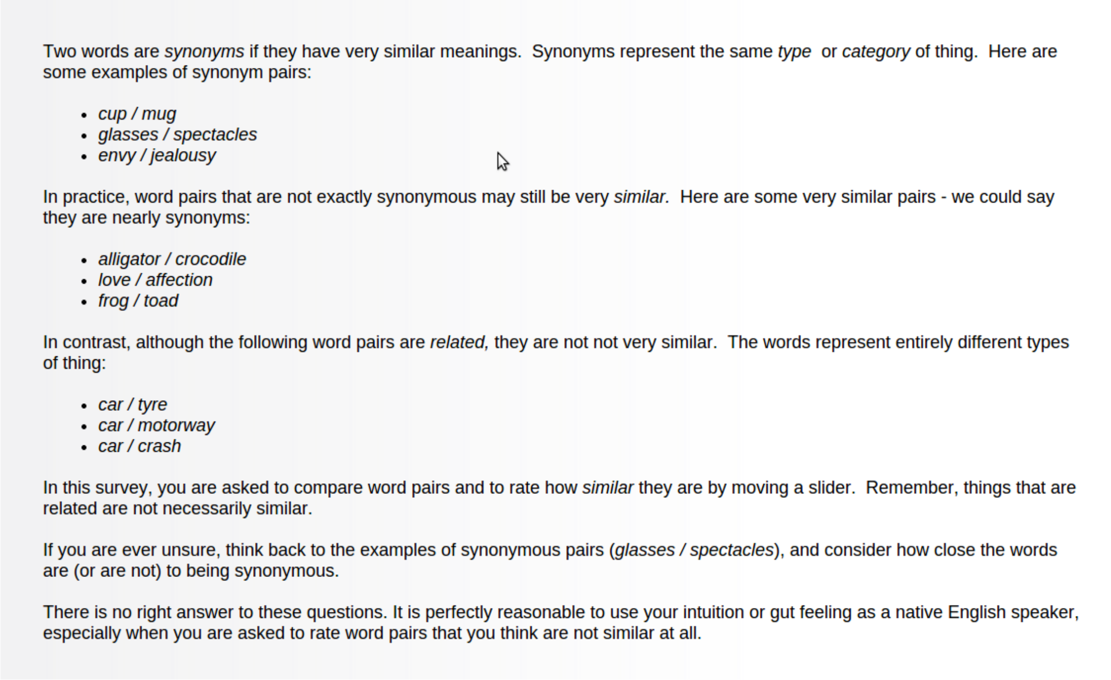
Table 3.1. Example of SimLex-999 similarity scores. A - adjective, N - noun, V - verb

The SimLex-999 also makes three other conceptual distinctions:

- **Concreteness:** Every concept in SimLex-999 is rated for its concreteness. SimLex-999 has a balanced set of concrete and abstract concept pairs.

- **Part-Of-Speech:** There is 111 adjective-adjective pairs, 666 noun-noun pairs and 222 verb-verb pairs in SimLex-999.
- **Free-Association:** SimLex-999 also contains independent empirical measure of strength of relatedness between the pairs.

Figure 3.1 shows the annotator instructions for SimLex-999. The instructions did not formalize the meaning of similarity. They did, however, explain its difference with association instead. It was preferred that the annotators used their intuition as a native speaker of the language.



Two words are *synonyms* if they have very similar meanings. Synonyms represent the same *type* or *category* of thing. Here are some examples of synonym pairs:

- *cup / mug*
- *glasses / spectacles*
- *envy / jealousy*

In practice, word pairs that are not exactly synonymous may still be very *similar*. Here are some very similar pairs - we could say they are nearly synonyms:

- *alligator / crocodile*
- *love / affection*
- *frog / toad*

In contrast, although the following word pairs are *related*, they are not not very similar. The words represent entirely different types of thing:

- *car / tyre*
- *car / motorway*
- *car / crash*

In this survey, you are asked to compare word pairs and to rate how *similar* they are by moving a slider. Remember, things that are related are not necessarily similar.

If you are ever unsure, think back to the examples of synonymous pairs (*glasses / spectacles*), and consider how close the words are (or are not) to being synonymous.

There is no right answer to these questions. It is perfectly reasonable to use your intuition or gut feeling as a native English speaker, especially when you are asked to rate word pairs that you think are not similar at all.

Figure 3.1. Instructions for SimLex-999 annotators. Image extracted from Hill et al. article [14].

In total, five hundred residents of the United States were recruited from Amazon Mechanical Turk. The participants were divided into groups to rate pairs, each word pair was rated by about 50 participants.

3.2 Translating the SimLex-999 Data Set

Translation of SimLex-999 into Estonian was done in three parts. First, it was automatically translated with an English-Estonian dictionary and the Google

Translation API. After that, all the pairs were manually checked and corrected by the author of this thesis, who is native speaker of Estonian. The final translation was agreed by a second person with a background in linguistics. She corrected 172 word pairs in total. To keep the translated data set as similar to the SimLex-999 as possible, only one word translations were used and the translations with the same POS as the English word was used.

There were some concerns with the translations.

One word translation variant wasn't always the best translation for the word. For example word *cooperate* was translated into *koopereeruma* even though *koostööd tegema* would have been a better translation, because it is more widely used.

It was not always possible to use the same translation of a word in every pair it occurred due to the fact that a word had a sense in English which the equivalent Estonian word did not possess. For example word *ball*, which occurred in pairs *ball - costume* and also *ball - basket*. It is obvious for English language, that in the first word pair *ball* is the formal occasion, where people dance, and in the second word pair, it takes the meaning of a round object, used in sporting activities. Translating this word into one word would lose one of the senses.

There were some situations where two different English words had the same translation in Estonian language. This is the opposite problem of the previous issue - Estonian words can have senses that English equivalent words do not possess. For example, words *north* and *bottom* can be both translated into Estonian word *põhi*. It was chosen to keep those translations because the alternative translations were not expressing the similarity with the other word very well.

There were some pairs, that were almost synonyms, and were represented with different words, but in Estonian language, there was only one word for these words. For example, words *taxi* and *cab*, there is no difference between those words in Estonian language, there is only one word - *takso*. For differentiation purposes, word *taksi*, which is not commonly used by native speakers, was used for the translation.

After all the checks were done, the data set was finalized and named EstSimLex-999.

3.3 Word Pair Scoring

Four native speakers of Estonian were asked to score all the 999 word pairs in EstSimLex-999 based on their similarity. A translated version of the SimLex-999 instructions [14] was given to the annotators prior to them starting their work. These translated instructions can be seen in the Appendix.

Instructions contained two main points for the annotators:

- words are similar if they are synonyms or nearly synonyms e.g. *toad - frog*
- words can be related, but not similar e.g. *car - highway*

After that, all the scores from annotators were checked, if there were some obvious differences between the scores, then they were asked to reconsider their answer. Table 3.2 shows some word pairs in English and Estonian and their assigned similarity scores.

word 1	**word 2**	**sõna 1**	**sõna 2**	**Pos**	**SimLex999**	**EstSimLex999**
old	new	vana	uus	A	1.58	0
crucial	important	ülioluline	tähtis	A	8.82	9.25
mouth	lip	suu	huul	N	7.10	7
chicken	rice	kana	riis	N	1	1.43
get	buy	saama	ostma	V	5.08	3.25

Table 3.2. Subset of SimLex-999 and EstSimLex-999 scores

The inter-annotator agreement was calculated to check the consistency of the annotations. As was done in the article by Hill et al. [14], the inter-annotator agreement was computed as the average pairwise Spearman ρ correlation between all the ratings. The overall agreement is ρ=0.77. This score cannot be directly compared with the SimLex-999 inter-annotator score (0.67) because the number of annotators is too different. Overall, this score shows that the annotators were capable of rating various concepts consistently and were able to understand the task in hand.

It can be seen in Figure 3.2 that agreement was not uniform across different concept types. Least per-pair variability is occurring within adjective subset.

EstSimLex-999 scores were also compared with the SimLex-999 scores. Spearman's correlation coefficient (ρ) was 0.83.

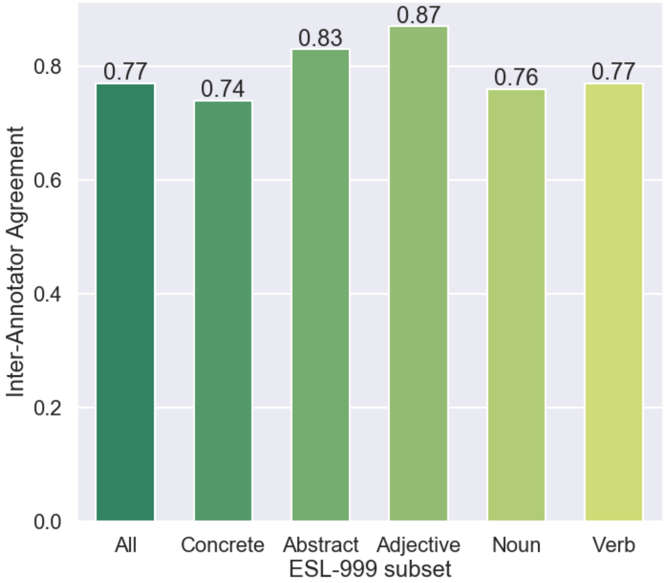


Figure 3.2. Inter-annotator agreement for ratings of concept types in EstSimLex-999 (ESL-999)

4 Evaluation of Computational Models

Three different computational models were tested to find how well can various models estimate similarity between word pairs compared with human annotation scores. Similarity scores from EstSimLex-999 and SimLex-999 were used for comparison to see how language can influence the similarity scores. All of these models are based on Estonian language resources: corpora, taxonomies and lexical ontologies with the exception of the computer vision models, which are all using image data.

The rest of this chapter describes the testing process of the computational models. All of the implemented methods for this task are written in Python 3.6 and can be accessed from the GitHub repository (see Appendix for more information). Pearson (r), Spearman (ρ) and Kendall Tau (τ) correlation coefficients were computed between similarity scores from the models and the similarity scores from SimLex-999 and EstSimLex-999.

4.1 Evaluation of Distributional models

As was said in the Background chapter about the distributional hypothesis that similar words tend to be together in a sentence. This hypothesis is put to a test in this thesis.

In total, 38 publicly available distributional models were evaluated on their performance on EstSimlex-999 and SimLex-999. If the embedding vector corresponding to a word in EstSimLex-999 was missing, all such pairs that contained that word were discarded. Similarity between embeddings was computed as the cosine similarity.

Eleri Aedmaa's models [4] contain 20 CBOW and 9 Skip-Gram models with different parameter settings trained on the lemmatized version of etTenTen: Corpus of the Estonian Web [27]. She also produced sense vectors, which are learned from the word embeddings using SenseGram software. She configured the number of dimensions, window size, minimum count threshold and number of iterations for the models. Her model names were in format: *architecture_ dimensions_ window_ minc_ iter*. Possible values for these parameters:

- architecture - CBOW or Skip-gram: *cbow*, *skip*
- dimensions - number of dimensions: *100, 150, 300, 450, 750*

[4] http://datadoi.ut.ee/handle/33/91

- window - window size: *5, 10, 15, 30*
- minc - minimum count threshold: *2, 5, 10, 15*
- iter - number of iterations: *5, 10, 20*

EstNLTK contains 8 pretrained word embeddings [28] trained with Word2Vec software. They use the Estonian Reference Corpus [16] for the training data. Half of the models are trained on the original and the other half on the lemmatized version of the corpus. The Estonian Reference Corpus contains about 1.3 billion words, which are mainly scraped from online newspaper publications.

Facebook research provides one CBOW model [10] trained on Estonian Wikipedia using fastText [3] software.

4.2 Evaluation of Semantic Networks

Two semantic networks: Estonian Wordnet and Estonian Wikipedia page and category taxonomy were used to find similarity between concepts mapped to EstSimLex-999 words. Three path based measures were calculated: path similarity, Leacock & Chodorow and Wu & Palmer for both of the networks.

4.2.1 Estonian Wordnet

Estonian Wordnet version 2.2 [5] was downloaded as an XML file. This Wordnet contains about 86000 synsets.

As there are many senses for one word in Estonian Wordnet, a disambiguation process is implemented to find the most probable sense. For that, the Cartesian product between the word senses is generated. This will produce many word-sense pairs, all the possible similarity scores are calculated using every possible sense. The word-sense pair that has the highest similarity is used. This way of mapping the sense to a word has been shown to be very effective, achieving 90 per cent precision for the Estonian WordNet[2]. For example, for the word *klaas*, there is 4 senses in Wordnet and for word *kristall*, there is 2 senses. Table 4.1 shows possible path similarity scores. In this example, *s-klaas-n1* and *s-kristall-n2* would be used because this pair yields the highest similarity.

[5] https://gitlab.keeleressursid.ee/avalik/data/blob/master/estwn/estwn-et-2.2.0.xml

	s-kristall-n1	s-kristall-n2
s-klaas-n1	0.25	0.5
s-klaas-n2	0.13	0.13
s-klaas-n3	0.08	0.08
s-klaas-n4	0	0

Table 4.1. Possible word-sense combination and their path similarities

4.2.2 Estonian Wikipedia

Wikipedia's page and category taxonomies were used to compute the similarity between the concepts corresponding to the words in EstSimLex-999. These taxonomies were extracted from Estonian Wikipedia by the language technology research group at Università Roma Tre [8]. In total, there are 96465 concepts in this Wikipedia page and category taxonomy.

The page taxonomy contains about 86000 concepts. For every page, there is one or many corresponding superordinate pages. All the path-based similarity measures were implemented by using these links between the Wikipedia pages. Due to a fact that there were many superordinate pages for a page, the most relevant path between the pages connected via these hypernym links had to be chosen. This was done with the same method as described previously. For example, in Figure 4.1 there are 3 different paths to page *Leib* to a root page and also 3 different paths from page *Jahu* to a root page. As can be seen from the figure, there can also be many root pages. In all cases, the path which gave the highest similarity was used.

The category taxonomy contains 13738 concepts. This taxonomy could not be used separately from the page taxonomy, as there were only 9 words in EstSimLex-999 that could be mapped to a Wikipedia category. Because of that, all the words were first mapped to a Wikipedia page and then it was switched to a category taxonomy using the categories linked to this page. Figure 4.2 shows a part of the page and category taxonomy for word *koer*. In this case, word *koer* has a Wikipedia page named *Koer*. This page has two Wikipedia categories linked to it - *Koer* and *Koerlased*. This also shows that page could have many categories linked to it. Here again, the paths that yield the highest similarity were used.

Only noun pairs were used from EstSimLex-999 because adjectives and verbs were not represented as a page or a category in Wikipedia.

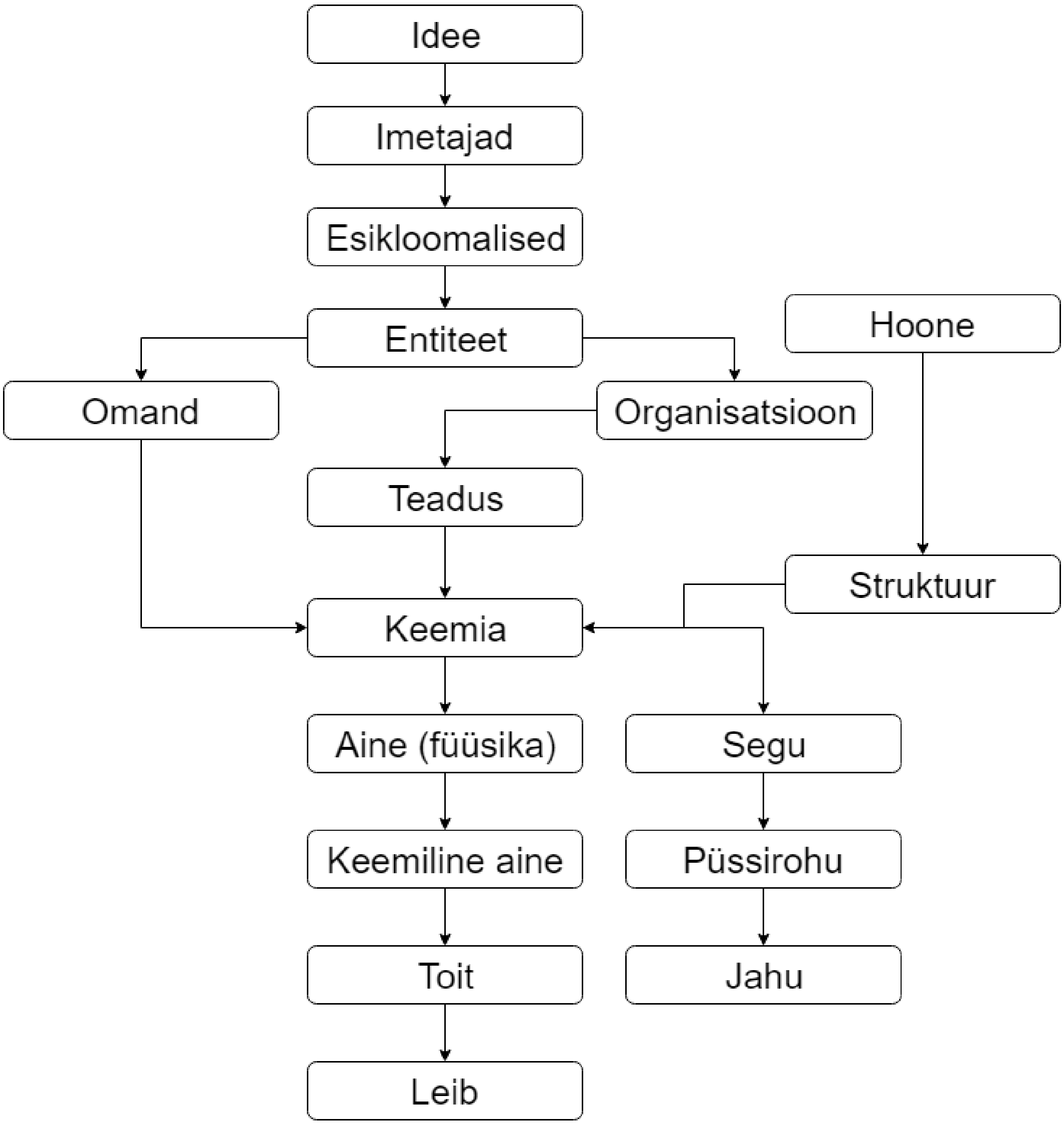


Figure 4.1. Part of page taxonomy

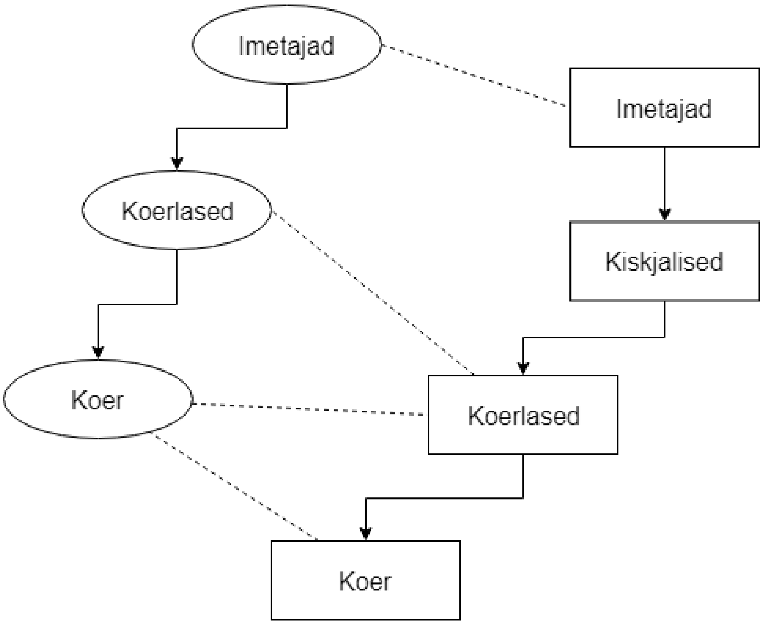


Figure 4.2. Part of page and category taxonomy. Circles represent Wikipedia pages and rectangles represent Wikipedia categories

4.3 Evaluation of Computer Vision Models

In this subsection, the process of the image collection and the used computer vision models are described.

4.3.1 Downloading Images

Images only for the concrete words were downloaded, abstract words are hard or even impossible to represent with a picture. For example, abstract word *beautiful* does not have any definite meaning and thus cannot have a specific image that could represent it. Only word pairs which consisted of words with concreteness score at least 4.8 were used. A total of 136 words met this requirement.

An image scraper was implemented for downloading images from Yandex Images [6]. About 200 images were initially downloaded for every word. Due to duplicates and images that didn't represent the word the actual amount was lower in the end.

Requirements for the downloaded images were:

[6] https://yandex.com/images/

- image should be in reasonable quality
- image should display one object only
- image should be in JPEG, JPG or PNG file format

4.3.2 Convolutional Autoencoder

The idea behind using autoencoders for similarity estimation is that, provided the dimensionality of the autoencoder is low enough, it will be forced to only learn the most important and meaningful features of the data during training. Presumably these features would be important for similarity estimation as well. The trained encoder can be used to embed the image into a lower-dimensional encoding vector and that vector can be used to compute the cosine similarity.

The Convolutional autoencoder's (CAE) encoder used here consists of 3 convolutional layers, each followed by a max-pooling layer, which reduce the dimensions of the outputs. The decoder consists of 3 convolutional layers, which are followed by upsampling layers.

The downloaded images described previously were used as a training data for the CAE. Training took about 8 hours to complete. Figure 4.3 shows some reconstructed images by the CAE. Upper images are the input images and the lower images are the reconstructed ones. After training, the encoder part was extracted from the CAE. This encoder was used to get a vector representation of the images. Similarity between images corresponding to a word in SimLex-999 was calculated as the cosine similarity between the vectors. The final similarity score for a word pair was average of all the assigned scores from the model between every word pair represented as images. For example, if there were 200 images for word *plane* and 200 images for word *airport*, then the model would have to calculate 40000 similarity scores between the words.

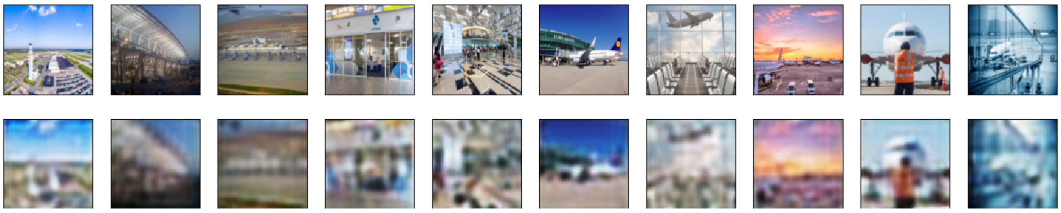


Figure 4.3. Example of the reconstructed images from the trained CAE model

4.3.3 Pretrained Convolutional Neural Network

The second computer vision model was a pretrained convolutional neural network, that won the ImageNet competition in 2015. This model is an architecture called Residual Network (ResNet) [12] invented by Microsoft Research. It has many variants with different layer sizes.

In this thesis ResNet-18 model with 18 residual layers was used. Architecture of ResNet-18 can be seen in Figure 4.4. As can be seen from the figure, ResNet-18 at first uses 7x7 convolution with stride 2 for downsampling the input. After that comes 8 residual blocks, each consisting 2 convolutional layers. The last layer is average pooling which creates 1000 feature maps and averages it for each feature map. Result from that is a 1000 dimensional vector which is fed to softmax layer.

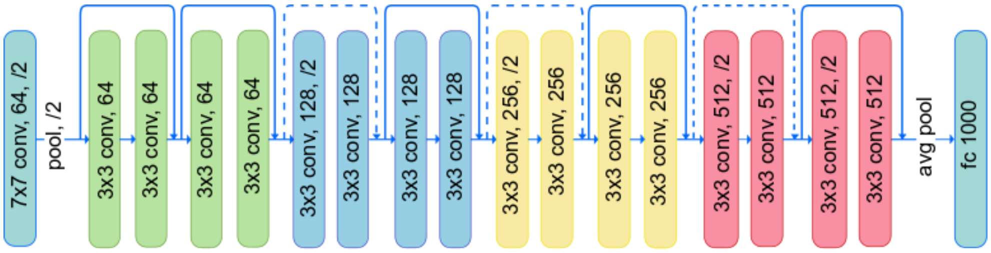


Figure 4.4. ResNet-18 architecture

As the model was pretrained, no training was needed. All the downloaded images were fed to the model, the representation of the image was read from the average pool of the final layer before prediction. The process was the same as was with the CAE model: the average of all the cosine similarity scores between word pairs from the filtered SimLex-999 was used as the final similarity score from the model.

5 Results

In this chapter, all the results are presented. First, results from distributional models are shown, then the results for semantic networks and finally results for computer vision models are shown. Pearson (r), Spearman (ρ) and Kendall (τ) correlations are calculated between the model similarity scores and SimLex-999 (SL-999) and EstSimLex-999 (ESL-999) similarity scores.

5.1 Results from Distributional models

On average, about 14 word pairs were discarded from each of the distributional models. Table 5.1 shows only the best results from the different sources. All the computed correlations can be seen from the Appendix.

Best model was Eleri Aedmaa's CBOW model [Reference the paper which presented the model again here] with dimension size 300 and window size 1. Spearman correlation coefficient was 0.42. Correlation coefficients between models' similarity scores and EstSimLex-999 human annotations are higher than between SimLex-999 human scores and models' similarity scores.

	SL-999			ESL-999		
	r	ρ	τ	r	ρ	τ
cbow_1	**.42**	.42	.29	.0.46	**.47**	.33
sg_2	.37	.36	.24	.41	.42	.3
cbow_3	.33	.33	.23	.33	.34	.24

Table 5.1. Best results from different sources. cbow_1 is Aedmaa's CBOW model with dimension size 300 and window size 1. sg_2 is EstNLTK Skip-gram model with dimension size 200 and window size 5. cbow_3 is Facebook research CBOW model with dimension size 300 and window size 5.

Sense vectors didn't perform better compared with the word vectors. SenseGram induced about 1.6 senses per word in EstSimLex-999. About 300 word pairs from EstSimLex-999 had more than one sense. All the results from the sense vectors can also be seen from the Appendix. The reasoning behind it was studied with comparing regular word vectors similarity estimates with sense vectors estimates. On average regular word vectors estimates were higher than sense vectors. Table 5.2 shows top 5 most different similarity estimates from sense and word vectors.

To see, where the distributional models predicted incorrectly, it was also explored the top 5 similarity scores, that were the most different from the EstSimLex-

word 1	word 2	sõna 1	sõna 2	Word vector	Sense vector
couple	pair	duo	paar	5.73	3.22
pupil	president	koolilaps	president	5.34	2.05
liquor	century	liköör	sajand	5.75	2.72
nice	cruel	tore	julm	6.2	3.81
whiskey	gin	viski	džinn	6.17	8.99

Table 5.2. Top 5 similarity scores, that were the most different from the regular vectors and sense vectors from cbow_1 model

999 human scores. Table 5.3 shows the top 5 word pairs that cbow_1 model predicted wrong. Model's predictions are scaled to the range 0-10 for better visual comparison.

word 1	word 2	sõna 1	sõna 2	SL-999	ESL-999	Model
short	long	lühike	pikk	1.23	0.5	8.29
smart	dumb	tark	rumal	0.55	0	7.29
dog	cat	koer	kass	1.75	1	8.97
leave	enter	lahkuma	sisenema	0.95	1.5	7.32
shrink	grow	kahanema	kasvama	0.23	0.5	7.69

Table 5.3. Top 5 similarity scores, that were the most different from the EstSimLex-999 and SimLex-999 human scores from cbow_1 model

Additionally, EstSimLex-999 was divided into three subsets: one, that contained only adjectives, one that contained only nouns and a third that contained only verbs. Figure 5.1 shows the average performance of distributional models on different part of speech subsets of the EstSimLex-999 word pairs. It was found that distributional models can find similarity on subset containing nouns better than on other subsets. Verbs pairs were the hardest for the models. Interestingly, models' similarity estimations for adjectives correlate with SimLex-999 similarity scores better than with EstSimLex-999 similarity scores.

EstSimLex-999 was also divided into two subsets: one concrete subset, containing 250 of most concrete word pairs from EstSimLex-999 and the other to abstract subset, containing 250 least concrete word pairs from EstSimLex-999. Figure 5.2 shows the average performance of distributional models on concrete and abstract subset of the EstSimLex999. Models can better estimate similarity on abstract subset of EstSimLex-999. Correlation coefficients are again higher for the EstSimLex-999 similarity set.

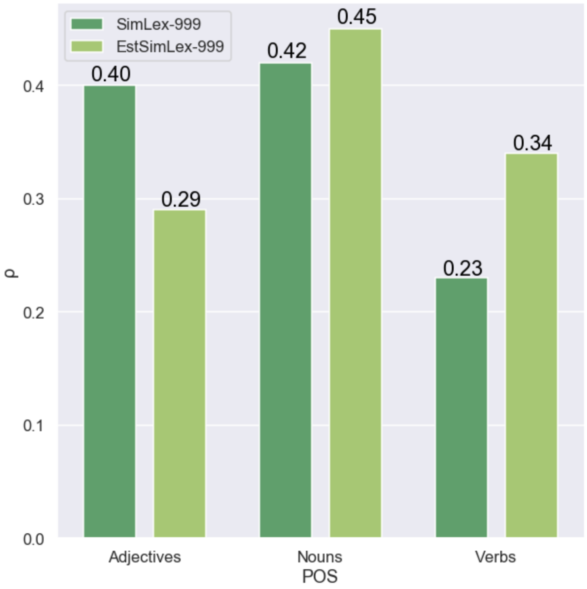


Figure 5.1. Average performance on POS-based subsets

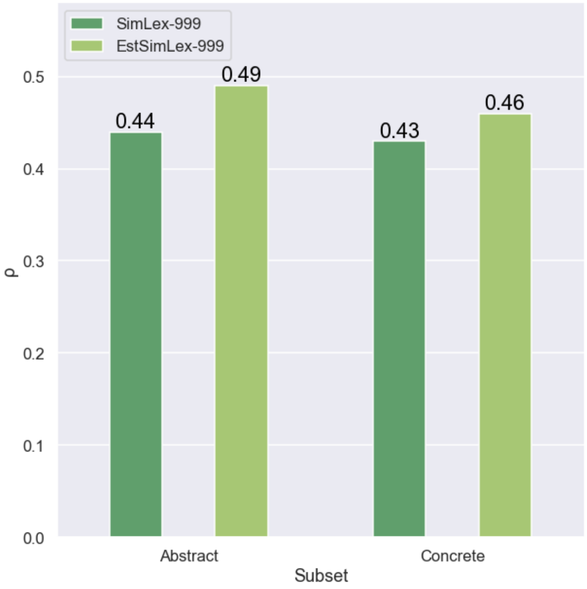


Figure 5.2. Average performance on POS-based subsets

5.2 Results from Semantic models

It was possible to map 770 word pairs onto the Estonian Wordnet. Table 5.4 shows all the results. Best was simple path similarity (PS) for both similarity sets. Wu & Palmer (WuP) similarity measure gave quite good correlation for EstSimLex-999 set as well, but not so good for SimLex-999 set.

	SL-999			**ESL-999**		
	r	ρ	τ	r	ρ	τ
PS	**.47**	**.47**	.35	**.54**	.52	.39
LC	.36	.36	.26	.41	.43	.31
WuP	.41	.45	.32	.49	.53	.39

Table 5.4. Results from Wordnet. PS - path similarity, LC- Leacock & Chodorow, WuP - Wu & Palmer

Results from Wikipedia page and category taxonomies were worse compared with the Wordnet results. When only the page taxonomy was used, it was possible to map 201 word pairs to a Wikipedia page and when using both: page and category taxonomies, it was possible to map 109 word pairs to a Wikipedia page. This count is lower, because even though these pages exists in the taxonomy, there was no path connecting those pages. Results from the page taxonomy can be seen from Table 5.5. Best results were obtained with Wu & Palmer similarity.

	SL-999			**ESL-999**		
	r	ρ	τ	r	ρ	τ
PS	.32	.31	.22	**.37**	.34	.24
LC	.31	.3	.21	.35	.34	.28
WuP	**.39**	**.37**	.28	.4	**.37**	.27

Table 5.5. Results from Wikipedia page taxonomy. PS - path similarity, LC- Leacock & Chodorow, WuP - Wu & Palmer

Results from using both taxonomies can be seen from Table 5.6.

5.3 Results from Computer Vision Models

Table 5.7 shows all the results from computer vision models. ResNet-18 performed better than the convolutional autoencoder model, achieving Spearman correlation coefficient 0.38 on both SimLex-999 and EstSimLex-999.

	SL-999			**ESL-999**		
	r	ρ	τ	r	ρ	τ
PS	**.29**	.28	.2	.22	**.24**	.18
LC	.26	.24	.17	.2	.15	.25
WuP	.12	.18	.13	.18	.19	.13

Table 5.6. Results from merged Wikipedia category and page taxonomy

Model	**SL-999**			**ESL-999**		
	r	ρ	τ	r	ρ	τ
CAE	.25	.28	.19	.17	.22	.15
RN18	.37	**.38**	.26	.34	**.38**	.27

Table 5.7. Results from auto-encoder (CAE) and ResNet-18 (RN18)

5.4 Discussion

Results from distributional models show that their similarity scores moderately [7] correlate with human annotated similarity scores. It can be seen from the models' similarity estimations, that they actually rate relatedness between words and not the underlying similarity. All such pairs, that the models' estimated as highly similar, were actually related and not similar.

Also, it can be said, that CBOW models are bit better for estimating human scores than Skip-gram ones. Differently from what we have expected the window size doesn't seem to affect the models' performance. The models with window size 1 and window size 30 were performing equally well. However because only 1 model for each configuration was tested, this needs more testing. The dimension size of the embeddings affects the results only a little, the models with more dimensions gave better results on average.

Sense vectors were performing worse than the regular word vectors, but not that much. Why such results occurred is not known and out of the scope of this thesis.

Analyzing the models' performance on different subset of EstSimLex-999, it can be said that distributional models perform on noun subset better than on adjectives and verbs. This is in contrast with a finding in [14], where results showed that the models were able to estimate similarity better on adjective word pairs. Also this in contrast to the fact that generally there are more nouns that are concrete, yet the results show that models were performing better on abstract subset of

[7] Correlation strength is usually considered moderate if the coefficient is in the range 0.4-0.59

EstSimLex-999.

Wikipedia taxonomies were underwhelming compared to Wordnet. Estonian Wikipedia is also really small compared with other Wikipedias. Many concepts are not defined as a page or a category, this makes these taxonomies less meaningful. Path-based similarity measures are heavily dependant on the quality of the graph. This means that manually built resources, which have clearly defined semantic relations are better predictors of human similarity.

The convolutional autoencoder was performing worse compared to ResNet-18. This was an expected result as ResNet-18 was trained on millions of images from ImageNet, but CAE model was only trained on about 25000 images. Overall, results from computer vision models are comparable with distributional models. In general, it can be said that even though computer vision models can extract similarity from images to some degree, there are still other factors, that affect similarity. There were words like bed and bedroom that the computer vision models rated as very similar even though the human score was low. This is reasonable because bed was in every bedroom picture, making these words visually similar.

In general, it could be said, that correlation between all the used computational models and EstSimLex-999 similarity scores are better than the correlation between models and SimLex-999 similarity scores. This means, that language indeed has an effect on similarity.

To conclude, semantic networks that are manually built based on the semantic relations were best at predicting the human annotated similarity scores.

6 Conclusion

In this thesis, three families of computational models for the Estonian language were evaluated for their ability to estimate the similarity between concepts. The goal of this work was not to obtain the best correlations between the models and human similarity scores, but to see how traditional models' similarity estimations correlate with human annotations. It was also studied if the language of the word pairs had any effect on human annotated scores. Lastly, it was studied if simple computer vision models alone can be enough to estimate the similarity between concepts.

To test these models, a human annotated data set was created for the Estonian language. This set contains 999 Estonian word pairs which are rated based on their similarity. This set was used for the evaluation of computational models.

It was found, that manually created resources like Wordnet are best for estimating similarity between concepts. Other tested models were not performing that well. It was also found that models estimations correlated with EstSimLex-999 human scores better than with SimLex-999 scores, which shows that language has a (slight) effect on similarity. Additionally, it was discovered that computer vision models can estimate similarity to some degree, though it does not explain all similarity.

In this thesis the contribution is threefold. First, a new resource with human annotated similarity scores was created for the Estonian language. Second, it was discovered how well can computational models for Estonian language estimate similarity and which one should be preferred for different natural language processing tasks. Thirdly, it was contributed to the research on how the computer vision models alone can capture similarity.

In conclusion, it can be said that all the goals of this thesis were met.

7 Acknowledgements

A number of persons have helped me with this work. I want to thank my supervisor Eduard Barbu for discussions, instructions and support. I would also like to thank Kadri Muishineck for recommending a good linguist, Aliis Aalmann. Special thanks to Aliis Aalmann for correcting and checking the EstSimLex-999 translations and Eleri Aedmaa for making available the Word2vec and sense vectors. I would like to thank the anonymous annotators for rating 999 Estonian word pairs and Al William Tammsaar for correcting my English. Last, but not least I would like to thank my cat for cheering me up during the work for this thesis.

References

[1] Vincent Andrearczyk and Paul F Whelan. Deep learning in texture analysis and its application to tissue image classification. In *Biomedical Texture Analysis*, pages 95–129. Elsevier, 2017.

[2] Eduard Barbu, Heili Orav, and Kadri Vare. Topic interpretation using wordnet. In Kadri Muischnek and Kaili Müürisep, editors, *Baltic HLT*, volume 307 of *Frontiers in Artificial Intelligence and Applications*, pages 9–17. IOS Press, 2018.

[3] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[4] Elia Bruni, Gemma Boleda, Marco Baroni, and Nam-Khanh Tran. Distributional semantics in technicolor. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 136–145. Association for Computational Linguistics, 2012.

[5] Elia Bruni, Giang Binh Tran, and Marco Baroni. Distributional semantics from text and images. In *Proceedings of the GEMS 2011 workshop on geometrical models of natural language semantics*, pages 22–32, 2011.

[6] Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. Placing search in context: The concept revisited. *ACM Transactions on information systems*, 20(1):116–131, 2002.

[7] J.R. Firth. *Papers in linguistics, 1934-1951.* Oxford University Press, 1957.

[8] Tiziano Flati, Daniele Vannella, Tommaso Pasini, and Roberto Navigli. Multiwibi: The multilingual wikipedia bitaxonomy project. *Artif. Intell.*, 241:66–102, 2016.

[9] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning.* MIT Press, 2016. `http://www.deeplearningbook.org`.

[10] Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[11] Sebastien Harispe. *Semantic Similarity from Natural Language and Ontology Analysis*, volume 39. 2008.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[13] Michael A Hedderich, Andrew Yates, Dietrich Klakow, and Gerard de Melo. Using multi-sense vector embeddings for reverse dictionaries. *arXiv preprint arXiv:1904.01451*, 2019.

[14] Felix Hill, Roi Reichart, and Anna Korhonen. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695, 2015.

[15] Dan Jurafsky and James H. Martin. *Speech and language processing : an introduction to natural language processing, computational linguistics, and speech recognition.* Pearson Prentice Hall, Upper Saddle River, N.J., 2009.

[16] Heiki-Jaan Kaalep, Kadri Muischnek, Kristel Uiboaed, and Kaarel Veskis. The estonian reference corpus: Its composition and morphology-aware user interface. 01 2010.

[17] Tara Larrue, Xiaoxu Meng, and Chang-Mu Han. Denoising videos with convolutional autoencoders a comparison of autoencoder architectures. 2018.

[18] Claudia Leacock and Martin Chodorow. Combining local context and wordnet similarity for word sense identification. *WordNet: An electronic lexical database*, 49(2):265–283, 1998.

[19] Yann LeCun et al. Generalization and network design strategies. In *Connectionism in perspective*, volume 19. Citeseer, 1989.

[20] Fritz Lehmann. Semantic networks. *Computers & Mathematics with Applications*, 23(2-5):1–50, 1992.

[21] Chee Wee Leong and Rada Mihalcea. Going beyond text: A hybrid image-text approach for measuring word relatedness. In *IJCNLP*, 2011.

[22] Ira Leviant and Roi Reichart. Separated by an un-common language: Towards judgment language informed vector space modeling. *arXiv preprint arXiv:1508.00106*, 2015.

[23] Jonathan Masci, Ueli Meier, Dan Cireşan, and Jürgen Schmidhuber. Stacked convolutional auto-encoders for hierarchical feature extraction. In *Proceedings of the 21th International Conference on Artificial Neural Networks - Volume Part I*, ICANN'11, pages 52–59, Berlin, Heidelberg, 2011. Springer-Verlag.

[24] Olena Medelyan, David Milne, Catherine Legg, and Ian H. Witten. Mining meaning from wikipedia. *International Journal of Human-Computer Studies*, 67(9):716 – 754, 2009.

[25] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[26] George Miller. *WordNet: An electronic lexical database*. MIT press, 1998.

[27] Kadri Muischnek. Eesti veeb 2013 (ettenten) korpus, morfoloogiliselt ühestatud. 2016. doi: `https://doi.org/10.15155/1-00-0000-0000-0000-0012el`.

[28] Siim Orasmaa, Timo Petmanson, Alexander Tkachenko, Sven Laur, and Heiki-Jaan Kaalep. Estnltk - nlp toolkit for estonian. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, may 2016. European Language Resources Association (ELRA).

[29] Kevin Patel and Pushpak Bhattacharyya. Towards lower bounds on number of dimensions for word embeddings. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 31–36, 2017.

[30] Maria Pelevina, Nikolay Arefiev, Chris Biemann, and Alexander Panchenko. Making sense of word embeddings. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 174–183, Berlin, Germany, August 2016. Association for Computational Linguistics.

[31] David C Plaut. Semantic and associative priming in a distributed attractor network. In *Proceedings of the 17th annual conference of the cognitive science society*, volume 17, pages 37–42. Pittsburgh, PA, 1995.

[32] R. Rada, H. Mili, E. Bicknell, and M. Blettner. Development and application of a metric on semantic nets. In *IEEE Transactions on Systems, Man and Cybernetics*, pages 17–30, 1989.

[33] Philip Resnik. Using information content to evaluate semantic similarity in a taxonomy. *arXiv preprint cmp-lg/9511007*, 1995.

[34] Zhibiao Wu and Martha Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 133–138. Association for Computational Linguistics, 1994.

[35] Torsten ZESCH and GUREVYCH Iryna. Wisdom of crowds versus wisdom of linguists-measuring the semantic relatedness of words. 2018. doi: 10.1017/S1351324909990167.

Appendix

I. Instructions for Annotators

This sections is showing the instructions, that were given to the EstSimLex-999 annotators to understand how to score word pairs based on the similarity. Instructions are translated version of the instructions that were given to SimLex-999 annotators.

Kaks sõna on sünonüümid, kui neil on väga sarnane tähendus. Sünonüümid tähistavad sama tüüpi või samas kategoorias olevaid asju.

Näiteid sünonüümi paaridest:

- tass / kruus
- ämber / pang
- ilus / kaunis

Sõnapaarid, mis ei ole küll sünonüümid, võivad siiski olla väga sarnased. Siin on mõned näied - võiks öelda, et nad on peaaegu sünonüümid:

- alligaator / krokodill
- konn / kärnkonn
- armastus / kiindumus

Kontrastiks, kuigi järgnevad sõnapaarid on seotud, ei ole nad sarnased. Need sõnapaarid esindavad täiesti erinevat tüüpi asju:

- auto / rehv
- auto / avarii
- auto / kiirtee

Järgenvas küsitluses palutakse Sul sõnapaare võrrelda ja hinnata, kui sarnased nad on skaalal 0-10. Jäta meelde, et asjad, mis on seotud, ei ole tingimata sarnased.

Kui juhtub, et oled ebakindel, mõtle taas näidis sõnapaaride peale (tass/kruus), ja kaalutle kui lähedal sõnad on olemaks sünonüümid.

Õiget vastus nendele küsimustele ei ole. Eesti keelt emakeelena kõnelejana on täiesti okei kasutada oma intuitsiooni või kõhutunnet, eriti kui arvad, et mõni sõnapaar ei ole üldse sarnane.

II. Results from the Distributional Models

	SL-999			**ESL-999**		
Model	r	ρ	τ	r	ρ	τ
cbow_100_5_10_20	0.37	0.36	0.24	0.37	0.39	0.27
cbow_150_15_10_20	0.39	0.39	0.27	0.40	0.43	0.30
cbow_150_15_5_20	0.39	0.39	0.27	0.41	0.43	0.30
cbow_150_5_10_20	0.37	0.36	0.25	0.38	0.40	0.28
cbow_150_5_10_5	0.37	0.37	0.25	0.39	0.41	0.28
cbow_150_5_5_20	0.37	0.36	0.25	0.38	0.40	0.28
cbow_300_10_10_5	0.40	0.40	0.27	0.42	0.44	0.31
cbow_300_15_10_20	0.40	0.40	0.28	0.42	0.45	0.31
cbow_300_15_10_5	0.40	0.40	0.28	0.42	0.45	0.32
cbow_300_1_10_20	**0.42**	**0.42**	0.29	0.46	**0.47**	0.33
cbow_300_30_10_20	0.40	0.41	0.28	0.43	0.46	0.32
cbow_300_5_10_10	0.39	0.38	0.27	0.41	0.43	0.30
cbow_300_5_10_20	0.38	0.37	0.26	0.40	0.42	0.29
cbow_300_5_10_5	0.39	0.39	0.27	0.41	0.43	0.30
cbow_300_5_15_5	0.40	0.40	0.27	0.42	0.44	0.31
cbow_300_5_2_20	0.38	0.38	0.26	0.40	0.42	0.29
cbow_300_5_5_20	0.38	0.38	0.26	0.40	0.42	0.29
cbow_300_5_5_5	0.39	0.38	0.26	0.41	0.43	0.30
cbow_450_5_10_5	0.39	0.39	0.27	0.41	0.44	0.30
cbow_750_5_10_20	0.39	0.38	0.27	0.41	0.42	0.30
skip_150_5_10_5	0.39	0.38	0.26	0.40	0.42	0.29
skip_300_10_10_5	0.38	0.38	0.26	0.40	0.42	0.30
skip_300_15_10_5	0.37	0.36	0.25	0.38	0.40	0.28
skip_300_5_10_10	0.39	0.39	0.27	0.42	0.44	0.31
skip_300_5_10_20	0.40	0.40	0.27	0.42	0.45	0.32
skip_300_5_10_5	0.39	0.39	0.27	0.42	0.45	0.31
skip_300_5_15_5	0.40	0.40	0.27	0.43	0.45	0.32
skip_300_5_5_5	0.40	0.40	0.27	0.43	0.45	0.31
skip_450_5_10_5	0.40	0.40	0.27	0.43	0.45	0.32

Table 7.1. Results from Eleri Aedma's word embeddings

	SL-999			**ESL-999**		
Model	r	ρ	τ	r	ρ	τ
cbow_100_5_10_20	0.34	0.33	0.22	0.33	0.36	0.25
cbow_150_15_10_20	0.35	0.35	0.24	0.37	**0.4**	0.27
cbow_150_15_5_20	0.36	0.35	0.24	0.37	0.39	0.27
cbow_150_5_10_20	0.34	0.34	0.23	0.35	0.37	0.26
cbow_150_5_10_5	0.33	0.32	0.22	0.33	0.35	0.24
cbow_150_5_5_20	0.34	0.33	0.22	0.34	0.36	0.25
cbow_300_10_10_5	0.34	0.34	0.23	0.35	0.38	0.26
cbow_300_15_10_20	0.36	0.36	0.25	0.37	**0.4**	0.27
cbow_300_15_10_5	0.35	0.34	0.23	0.36	0.39	0.26
cbow_300_1_10_20	**0.38**	0.37	0.25	0.36	0.39	0.27
cbow_300_30_10_20	0.36	0.36	0.24	0.37	**0.4**	0.28
cbow_300_5_10_10	0.34	0.33	0.23	0.34	0.37	0.25
cbow_300_5_10_20	0.34	0.33	0.23	0.34	0.37	0.26
cbow_300_5_10_5	0.32	0.32	0.21	0.33	0.36	0.25
cbow_300_5_15_5	0.32	0.32	0.22	0.33	0.36	0.24
cbow_300_5_2_20	0.33	0.33	0.22	0.34	0.36	0.25
cbow_300_5_5_20	0.34	0.33	0.23	0.34	0.36	0.25
cbow_300_5_5_5	0.32	0.31	0.21	0.32	0.34	0.23
cbow_450_5_10_5	0.34	0.34	0.23	0.34	0.37	0.25
cbow_750_5_10_20	0.35	0.35	0.24	0.36	0.39	0.27
skip_150_5_10_5	0.32	0.3	0.2	0.3	0.32	0.22
skip_300_10_10_5	0.31	0.3	0.21	0.3	0.33	0.23
skip_300_15_10_5	0.31	0.3	0.2	0.28	0.31	0.21
skip_300_5_10_10	0.34	0.33	0.23	0.34	0.37	0.25
skip_300_5_10_20	0.36	0.35	0.24	0.36	0.39	0.27
skip_300_5_10_5	0.33	0.31	0.21	0.32	0.34	0.23
skip_300_5_15_5	0.35	0.34	0.23	0.33	0.36	0.25
skip_300_5_5_5	0.34	0.32	0.22	0.31	0.33	0.23
skip_450_5_10_5	0.33	0.32	0.22	0.32	0.34	0.23

Table 7.2. Results from sense vectors

	SL-999			**ESL-999**		
Model	r	ρ	τ	r	ρ	τ
lemma_est_model_cbow100	0.32	0.31	0.21	0.34	0.36	0.25
lemma_est_model_sg100	0.35	0.33	0.23	0.38	0.39	0.28
lemma_est_model_cbow200	0.34	0.33	0.23	0.37	0.38	0.27
lemma_est_model_sg200	**0.37**	0.36	0.24	0.41	**0.42**	0.30
word_est_model_sg100	0.31	0.30	0.20	0.36	0.38	0.26
word_est_model_cbow200	0.29	0.27	0.19	0.34	0.35	0.25
word_est_model_sg200	0.32	0.31	0.21	0.37	0.38	0.27
word_est_model_cbow100	0.29	0.27	0.19	0.34	0.35	0.25
wiki_model_est_fastText	**0.33**	**0.33**	0.23	0.33	**0.34**	0.24

Table 7.3. Results from EstNLTK and Facebook research word embeddings

III. Code

Code used in this thesis and created EstSimLex-999 data set can be accessed from public GitHub repository [8].

[8] https://github.com/diffusa/SimLex-999-est-eng

II. Licence

Non-exclusive licence to reproduce thesis and make thesis public

I, **Claudia Kittask**,

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

   Computational Models of Concept Similarity for Estonian Language,

   supervised by Eduard Barbu.

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Claudia Kittask

10.05.2019