MSc Natural Language Processing – 2020 - 2021
UE 705 – Supervised Project

Understanding Morphological Analogies through a Semantic Approach

Students:
Safa Alsaidi
Amandine Decker
Puthineath Lay

Supervisors:
Miguel Couceiro
Esteban Marquer
Reviewer:
Maxime Amblard

December 18, 2020

Contents

Introduction **2**

1 Analogical Proportions and Equations **3**

1.1 Definition and Properties of Analogical Proportions 3

1.2 Solving Analogical Equations 4

1.3 Types of Analogies 5

1.4 Semantic Analogies and the Similarity of Semantic Relations 5

1.5 Morphological Analogy 6

2 Approaches to Solve Analogies - Semantics **8**

2.1 Textual Analogy Parsing 8

2.2 Dependency Relations 9

2.3 Turney's Vector Space Model 10

2.4 Neural Network Approach 11

2.4.1 Classification Task 11

2.4.2 Regression Problem 12

3 Approaches to Solve Analogies - Morphology **14**

3.1 Description of the Shared Task 14

3.2 Dataset 15

3.3 Kolmogorov Complexity 16

4 Conclusion and Future Work **18**

Bibliography **18**

Introduction

Understanding language is a complicated task. The major subfields of linguistics are the phonological, morphological, syntactic, semantic, and the pragmatic levels. The phonological level deals with individual speech sounds and the structure of phonemes. The morphological level deals with understanding how words are formed and describes how morphemes merge to form words. Morphemes are defined as the smallest unit of words that can not be divided any further. The syntactic level deals with understanding how sentences and phrases are structured. Syntax determines the order of words in a sentence. Each language has underlying syntax rules. These rules together with morphological rules make up a language's grammar rules. The semantic level deals with the meaning of the words and sentences. Semantics include all the meanings that a word can have. The pragmatic level is the context that influences the meaning of words or sentences. Some words have different meanings based on the context of the sentence they are used in.

For our project, we are interested particularly with morphology in the domain of analogies. In recent years, morphology has gained more prominence in the domain of analogies. Through analogies we will be able to model the relation between words and how they work together. That is why, for our project, we will analyze analogies by focusing on morphological word variations and how they determine the relation between different words. We try to work on SIGMORPHON dataset (Cotterell et al., 2016) to analyze how morphology could help us associate and construct valid word analogies. For this project, we will be using semantic approaches to solve morphological analogies. Our approach is different from the current state of the art approaches to solve morphological analogies.

In this report, we will start by a brief survey on analogical proportions and their types. The second chapter introduces the most common semantic approaches to solve analogical proportions. In contrast, the third chapter provides more insight to the main approach of our project and introduces the SIGMORPHON dataset that we will be using. In the last chapter, we will introduce our envision on our project and what we will be working on.

Chapter 1

Analogical Proportions and Equations

Analogy is defined as a method of reasoning. An analogy or a verbal analogy consists usually of four objects or words A, B, C and D and draws a parallel between the relation between A and B and the one between C and D. It can be expressed by what is called an analogical proportion which is a statement such as "A is to B as C is to D". Analogies can have different significations or types based on the objects or concepts. The next subsections give more insight into the different types of analogies.

1.1 Definition and Properties of Analogical Proportions

Analogical proportions are statements of the form "A is to B as C is to D". They express the fact that A should differ from B as C differs from D. These quaternary relations, usually written as follows : $A : B :: C : D$, obey the following axioms (Lepage, 2003):

1. $A : B :: A : B$ (reflexivity);
2. $A : B :: C : D \rightarrow C : D :: A : B$ (symmetry);
3. $A : B :: C : D \rightarrow A : C :: B : D$ (central permutation);
4. $A : B :: A : X \rightarrow X = B$ (unicity).

If unicity is not fulfilled then a proportion such as $A : B :: A : C$ with $B \neq C$ exists. Following axiom 3, $A : A :: B : C$ also holds. This proportion seems unreasonable since B and C are different while the elements of the first pair are the same.

Other properties and permutations can be inferred from these axioms such as $A : A :: B : B$ (identity), $A : B :: C : D \rightarrow B : A :: D : C$ (inside pair reversing) and $A : B :: C : D \rightarrow D : B :: C : A$ (extreme permutation).

1.2 Solving Analogical Equations

An analogical proportion becomes an equation if one of its four objects is unknown (Miclet et al., 2008). For example, the analogical equation of the analogical proportion "an apple is to tree as apples is to x" would be expressed as follows:

$$R = \{x | \text{apple is to tree as apples is to } x\}$$

. Solving this form of equations can be done by calculating the set R of sequences X which satisfy the analogy (Miclet et al., 2008). In this case, the observed sequence is based on morphological variations of two words: apple and tree.

The analogy "A is to B as C is to D" when expressed as an analogical equation is written as "$A : B :: C : D$." As we mentioned before, solving analogical equations is carried out when one of the 4 values is missing. The equation would, therefore, be expressed as "$A : B :: C : X$" and solving this equation consists in determining the value of X. To solve the equation, it requires the satisfaction of two axioms with two other equations (Delhay & Miclet, 2004):

1. $C : D :: A : B$ (symmetry of the 'as' relation);
2. $A : C :: B : D$ (exchange of the means);
3. $B : A :: D : C$ (inversion of ratios);
4. $D : B :: C : A$ (exchange of the extremes);
5. $D : C :: B : A$ (symmetry of reading);
6. $B : D :: A : C$ (symmetry of reading);
7. $C : A :: D : B$ (symmetry of reading).

Another axiom introduced is *determinism* (Delhay & Miclet, 2004). This axiom states that one of the equations mentioned below should have a unique solution and the other should be a consequence:

1. $A : A :: B : X \Rightarrow X = B$;
2. $A : B :: A : X \Rightarrow X = B$.

By taking into account the axioms of analogy introduced in this section, we can find the solution to different analogical equations. In addition to various applications as inference mechanisms, analogical proportions were also applied to classification tasks (Hug et al., 2016) which subsume analogical extension of training sets (Couceiro et al., 2017).

1.3 Types of Analogies

Analogies are classified and grouped based on the type of relation that exist between word pairs. The first implementation of proportions was introduced by Ancient Greeks and was used in the domain of numbers. Two examples worth mentioning are arithmetic proportion and geometric proportion (Couceiro et al., 2017). These two examples illustrate the analogical proportion statement of "A is to B as C is to D."

1. A, B, C, and D are proportional if $A - B = C - D$ (arthimetic);
2. A, B, C, and D are proportional if $\frac{A}{B} = \frac{C}{D}$ (geometric).

Other types of analogies include *semantic*, *classification*, *association*, and *logical/mathematical* (Betrand, 2016). Though each type is made of 2 wordpairs, the main difference exists in the form of relation between each of these words (Fibonicci, 2019).

Classification analogy is built on the concepts of objects and groups that these objects belong to (Fibonicci, 2019). Known examples are found in terms of animals and the kingdoms they belong to or utilities and the place they are found in. Association analogy breaks down the relationship between two objects or word pairs. The most used types of association analogies are object to characteristic, cause and effect, function, and sequential order (Dingyi, 1985). Association analogy is expressed as follows: "$A : B :: C : D$", which is read as "A is to B as C is to D". The relation between terms "A and B", and "C and D" are described to be equivalent to one another (Dingyi, 1985). The mathematical or logical analogy is based on the idea of solving basic mathematical problems that are written in an analogical form (Betrand, 2016). To solve such forms of analogies, the individual is expected to spot the relation between each of the problem pairs. Here are examples of the above-mentioned types of analogies:

- *Eagle*:*Bird*::*Tuna*:*Fish* (Classification)
- *Tornado*:*Destruction*::*Hurricane*:*Flood* (Association)
- $1 : 2 :: 2 : 4$ (Mathemathical)

Semantic analogies deal with the intended meaning of words included. Through reasoning, semantic analogies aim to finding the similar features and the common relationships that exist between word pairs (Schiff et al., 2009). Such types of analogies are found by identifying similarities between situations, inference making, learning new abstractions, and creating conceptual change (Schiff et al., 2009).

1.4 Semantic Analogies and the Similarity of Semantic Relations

Most articles on the analysis of semantic analogies mentioned two particular kinds of similarity: relational similarity and attributional similarity. Relational similarity is defined

as the similarity that exists in relations between objects and takes two or more arguments (*e.g.*, S collides with V, S is larger than V). In contrast, attributional similarity exists between attributes to state properties of objects and takes one argument (*e.g.*, S is green, S is small) (Medin et al., 1990). The term "synonyms" was derived as a result of the existence of a high degree of attributional similarity between word pairs, but if a word pair has a high degree of relational similarity, its relation is described as rather analogous. Semantic analogies are often represented in the form of $A : B :: C : D$. An example by (Daganzo, 1994) of relational similarity is between the word pair *traffic* : *street* and the word pair *water* : *riverbed*. In comparison, analogies such as *mason* : *stone* :: *carpenter* : *wood* are attributionally similar. Nonetheless, an undeniable relation exists between attributional and relational similarities. To simplify it, if there is a relational similarity between $A : B$ and $C : D$, then there is attributional similarity between $A : B$ and $C : D$ (Turney, 2006).

Due to the wide applications of attributional similarities in various problems, many algorithms have been proposed to measure attributional similarities between words. Some of those algorithms were used to solve problems such as recognizing synonyms (Landauer & Dumais, 1997), retrieving information (Deerwester et al., 1990), determining semantic relations (Turney, 2002), word sense disambiguation (Lesk, 1986), *etc.* Three main approaches to measure attributional similarities were highlighted in several articles: lexicon-based approach (Lesk, 1986; Budanitsky & Hirst, 2001; Banerjee & Pedersen, 2003), corpus-based approach (Lesk, 1986; Landauer & Dumais, 1997; Lin, 1998; Turney, 2001), or a mix of the two approaches (Resnik, 1995; Jiang & Conrath, 1997; Turney, 2006). Lexicon based is beneficial when we are trying to distinguish synonyms, one such example is WordNet; whereas, corpus-based approach depends on the context to determine word senses and uses grammatical collocations to describe each word in a pair (Turney, 2006).

1.5 Morphological Analogy

Recently, some research showed that analogical learning based on formal analogy can be applied to many problems in computational linguistics (Miclet & Delhay, 2003). To quote Haspelmath (Haspelmath, 2002) : "Morphology is the study of systematic co-variation in the form and meaning of words." When analyzing analogy in a morphological approach, we are looking at the co-variation in the form of a single word. For example, "reader is to doer as reading is to doing" is an analogy made of four tuples that present the different variations of the lexicons "read" and "do" (Miclet & Delhay, 2003). Analyzing analogies based on morphology allows the linguist to find the sequence of how the word could vary based on gender, plurality, tense, mood, *etc.* It also allows the linguist to predict how words change form based on these classified patterns even if he/she is not familiar with certain words.

The most common form of morphology in words is affixes. Most affixes can not stand alone and are therefore referred to as bound morphemes as they need other morphemes to be connected to. Affixes include prefixes, suffixes, infixes, and circumfixes (Krott et al., 01 Mar. 2001). Each of these groups differs in the position where the morpheme links to

a lexicon or to another morpheme. There are two categories of affixes that most articles refer to when analyzing analogies: derivational and inflectional (Krott et al., 01 Mar. 2001). Derivational affixes are made up by adding a morpheme to create a new word that may or may not still belong to the same part of speech so if was a verb, this added morpheme would make it a noun (Krott et al., 01 Mar. 2001). Inflectional morphemes are made up of morphemes that when added to the word change the grammatical feature (Lim et al., 2019). The most known inflectional morphemes depend on changing words for only grammatical reasons by adding morphemes to either change plurality/singularity, tense (past, present, *etc.*), comparative/superlative, *etc.*

In this project, we are interested in inflectional affixes in morphology to analyze analogies since through this variations we can determine a sequence rule. As a result, we will be able to understand the differences associated with each lexicon.

Chapter 2

Approaches to Solve Analogies - Semantics

Based on the project objective and the researchers' approaches, different models have been proposed to solve analogies. The models that we most frequently encountered were Textual Analogy Parsing approach, Dependency Relation approach, Vector Space Model, and Neural Network approach.

2.1 Textual Analogy Parsing

Textual Analogy Parsing (TAP) aims to extracting analogies from natural language. For instance to get the meaning of the sentence "According to the U.S. Census, whereas only 10% of White Americans live at or below the poverty line today, 28% of African Americans do.", one must recognize the comparison between White Americans and African Americans regarding the poverty line.

(Lamm et al., 2018b) propose a model to build analogy frames based on the Quantitative Semantic Role Labeling (QSRL) framework (Lamm et al., 2018a). A frame is a representation of a sentence where each span of text is associated with its semantic role. Analogy frames highlight shared content and compared content of sentences containing analogies.

FIGURE 2.1 describes the process of analogy frame building. Given a sentence, meaningful spans of text are identified (highlighted pieces of text) and mapped with their semantic role (such as SOURCE, TIME or VALUE). An analogy graph is then built, it represents the relations between the analogous facts of the sentence. The vertices of the graph are the spans of text identified and the edges the relations between these spans, relations can be FACT, EQUIVALENCE or ANALOGY.

The spans are selected thanks to a neural network. Given a sentence, each of its token is embedded with fixed words embeddings and linguistic features determined by CoreNLP (Pennington et al., 2014) are concatenated. These embeddings are passed through a neural network to create context-sensitive words embeddings. Then a conditional random field (CRF) (Lafferty et al., 2001) predicts the semantic role of each token. The spans are

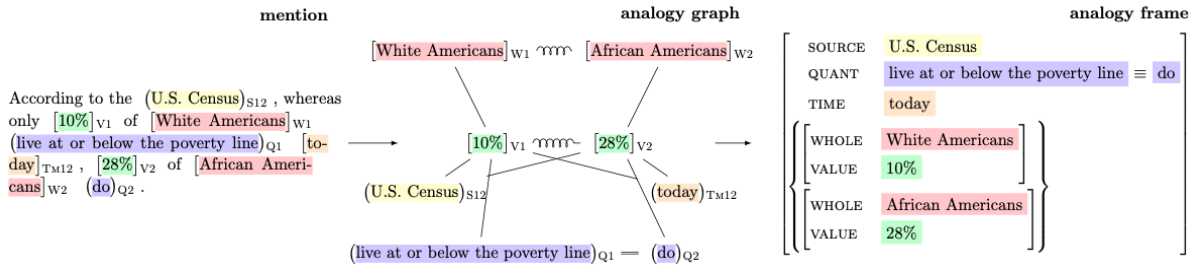

Figure 2.1: Analogy frame building process (Lamm et al., 2018b)

eventually created by merging contiguous tokens with the same role label.

Span and edge embeddings are then constructed based on the identified spans and their features. The graph is produced by decoding these embeddings : a role is attributed to the relevant spans and edges. The optimal decoder discussed in the article is based on integer-linear programming (ILP) (Roth & Yih, 2004; Do et al., 2012): the constraints of TAP are encoded and the decoder tries to find the optimal solution regarding these constraints.

2.2 Dependency Relations

A type of analogy used frequently in various NLP tasks is *lexical analogy* (Chiu et al., 2007). Similar to formal analogies, lexical analogies are also made of 4 word tuples. But for analogies to be considered lexical, the wordpairs should be semantically related. The semantic relations that exist between those word pairs are classified to be relational similarity; in other words, relation dependent. Lexical analogies have been widely applied in word sense disambiguation, information extraction, question-answering, and semantic relation classification (Chiu et al., 2007).

Developed by Andy Chiu, Pascal Poupart, and Chrysanne DiMarco, the approach introduced in this section "aims to find and generate lexical analogies from raw text data." (Chiu et al., 2007). This system uses dependency relations (DP) to classify word-pairs of semantic relations and compares those results with two machine learning algorithms: LRA (Latent Relational Analyzis) and the SGT (Similarity Graph Traversal) (Chiu et al., 2007). The method used was divided into two key problems: data extraction (identifying semantically related word pairs) and relation-matching (involves constructing lexical analogies by matching word-pairs of similar features). Relation-matching measures the similarity of the underlying relationship of the four words in an analogical proportion; it is the cosine measure of the corresponding vectors (Chiu et al., 2007). The system used for data extraction allows more than just SVO (subject-verb-object) extraction, where it can create dependency paths between the verb and the second nouns (Chiu et al., 2007). A dependency pattern based on the word-pair's dependency paths was established and used as a feature for the extracted word-pairs. The SGT system works on the notion of transitivity, hence not all relational similarities are transitive. The SGT system can be explained as such: if *word* 1 relates to *word* 2 and *word* 2 relates to *word* 3; this should mean that *word* 1 is related to *word* 3. But as stated previously, this is not the case for

all relational similarities; therefore, this algorithm is useful but limited.

2.3 Turney's Vector Space Model

The Vector Space Model (VSM) is a general model used in various applications in natural language processing and information retrieval to measure text similarity. (Turney, 2006) introduced the approach of the VSM of information retrieval to solve verbal analogies. The VSM approach has also been used to measure the semantic similarity of words (Lesk, 1969; Ruge, 1992; Pantel & Lin, 2002). Through this model, a vector of numbers was used to represent the semantic relations between word pairs. The VSM represents documents in a multidimensional space where each term is a dimension. The terms could be the words that appear in documents and the documents are linear combinations of vectors along the axes. Document similarity is used in information retrieval to determine which document is more similar to a given query. Queries and documents are represented in the same space.

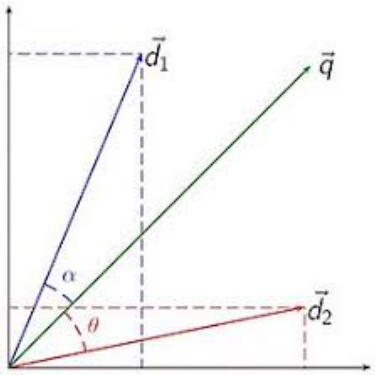

Figure 2.2: VSM with 2 documents and 1 query

FIGURE 2.2 illustrates a model of a two-dimensional vector space, where each dimension represents a term or object in the documents. Each document and query is presented as a point in space. In FIGURE 2.2, we have the query Q and we want to determine whether d_1 or d_2 is a better match to the query. For that, we use the angle of the vectors to present their similarity. In the figure, the similarity between d_1 and Q is proportional to the angle alpha and the similarity between d_2 and Q is proportional to the angle theta. In a better version, this is the cosine of alpha versus cosine of theta and in this quadrant they are in the same direction so if cosine is smaller that means the angle is also smaller, which also means that the similarity is larger. When it comes to analogies, the cosine of the angle was measured between the vector that represents $A : B$ and $C : D$ to determine the similarity between two word pairs in a four-tuple word analogy (Turney, 2006). (Turney, 2006) defined and recognized these analogies based on the standard practice of information retrieval: precision, recall, and F1 scores.

- Precision score is defined as the ratio of relevant documents retrieved over the total

number of documents. Formally:

$$precision = \frac{\text{number of relevant documents retrieved}}{\text{number of documents}}$$

- Recall score is defined as the ratio of relevant documents to the query that are successfully retrieved. Formally:

$$recall = \frac{\text{number of relevant documents retrieved}}{\text{number of relevant documents}}$$

- F1 score is defined as the ratio of the product of the precision and recall multiplied by 2 over the sum of precision and recall. Formally:

$$F1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

One of the recent machine learning methods in vector space that has been achieving considerable success in natural language processing is word2vec (Chen et al., 2017). It adapts a parallelogram model of analogy, which was first proposed by Rumelhart and Abrahamson (Rumelhart & Abrahamson, 1973). If the difference of the vectors is similar for two word pairs, they are described to be relationally similar (Chen et al., 2017).

2.4 Neural Network Approach

Neural networks are models that work similar to the human brain. They are commonly used in machine learning to solve various problems such as classification and regression. One interesting feature is that neural network models are learned from data without much prior knowledge (Kaveeta & Lepage, 2016). Here we will explore neural networks to classify and solve (word) analogies and analogical equations.

2.4.1 Classification Task

Recently, word embeddings are used to convert words to numerical vectors. They are n-dimensional vectors that try to detect meaning of the word and context in their values. For example, if S is the target vector space and W is the corpus of words, it is denoted embed(W) the subset of S standing for the words of W (Lim et al., 2019). Lim et al. (2019) propose a convolutional neural network (CNN) (see FIGURE 2.3) to classify the valid and invalid analogies[1]. They used GloVe embeddings to represent words in an n dimensional space. Since analogies are quaternary relations, they stack together the 4 vectors into an image of size $n \times 4$. The desired classifier should indicate that *engine*:*car*::*heart*:*human* is a valid analogy, whereas *car*:*engine*::*heart*:*human* is not. For training, the authors used the Google dataset (questions-words) containing $19,544$ analogies, each of which involving 4 distinct words. The structure of the proposed CNN together with the number of filters[2] is shown in FIGURE 2.3.

[1]We take a human-centered approach of valid analogy, and view analogies as being either true or false.

[2]Filters are relatively small matrices which produce the number of output channels (Mandy, 2019) and the number of the filters need to be set in each layer.

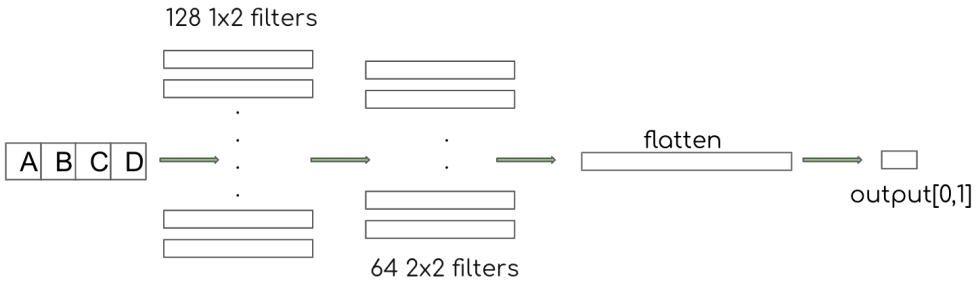

Figure 2.3: CNN classifier structure (Lim et al., 2019).

FIGURE 2.3 illustrates the first layer which has 128 filters of size $h \times w = 1 \times 2$ with strides [3] $(1, 2)$ and Regularized Linear Unit (ReLU) activation. The second layer has 64 filters of size $(2, 2)$ with strides $(2, 2)$ and ReLU activation. On the other hand, the third layer has one output and sigmoid activation as we want a score between 0 and 1(the output to show the result is true or false).

As a result, it is shown that it provides accuracy higher than 94% with CNN classifier.

2.4.2 Regression Problem

In classification task, (Lim et al., 2019) aim to classify word-analogies as valid or not. For the regression problem, the authors aim to solve analogical equations to find the 4th word by inputting only 3 words in quadruple of words. For example, they want to find the result of X from the analogical equation $A : B :: C : X$. The state of the art approach of looking for that X is cosine similarity multiplication (3CosMul) based on (Levy & Goldberg, 2014). (Levy & Goldberg, 2014) define 3CosMul is to find the similarity of the words; in addition to this, (Lim et al., 2019) decided to use a deep learning approach to look for the target output. In (Lim et al., 2019)'s experiment, they compared the result of using either 3CosMul or neural network for regression (the second model for regression) to see the best result.

In relation to this equation solving problem, there are 3 inputs (A, B, C) and 1 output (X). Because of this, it can be written as a function f such that $f_(A, B, C) = X$. They received not-so-good results from 3CosMul approach compared to the result of neural network approach. Presumably, it is because of 3CosMul does not refer to the similarities and dissimilarities between A and B on one side, and between A and C on the other side as shown in FIGURE 2.4. More precisely, we can take a look at functions as illustrated in FIGURE 2.4 with the description below:

1. function 1 (f_1) described a and b by the hidden link;
2. function 2 (f_2) described a and c by the hidden link;

[3]The stride is a parameter of the neural network's filter that adjust the amount of movement of the image, *e.g.*, if a neural network's stride is set to 2, then the filter will move 2 units at one time (Prabhu, 2018))

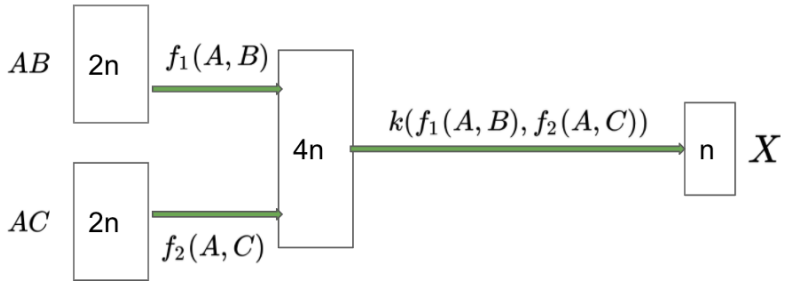

Figure 2.4: Neural network for regression structure (Lim et al., 2019).

3. function k is the last solution which combines the function $f_1(A, B)$ and $f_2(A, C)$, denoted as $X = k(f_1(A, B), f_2(A, C))$.

The function k is obtained from two input values: the output of f_1 and the output of f_2. The output of $k(f_1, f_2)$ is X, so $X = k(f_1(A, B), f_2(A, C))$. Since the target term of the network is X, so the word nearest to X is needed to be found, *e.g.:* the nearest neighbor of X should be the right answer in embed(W) (W is previously mentioned in classification part).

To emphasize, A, B and C are words embedding of dimension $n \in \{50, 100, 200, 300\}$. It can be written as $A, B, C \in \mathbb{R}^n$. As a consequence, the best overall accuracy result of neural network for regression is 79% at 100 dimensions, while the best overall performance of 3CosMul is 68.1% at the 300 dimensions. All in all, the neural network regression seems to perform better than 3CosMul.

Chapter 3

Approaches to Solve Analogies - Morphology

3.1 Description of the Shared Task

In our project, we aim to adapt the framework of the semantic approaches and try to apply them to solve morphological analogies. Therefore, we chose to base our project on the shared task from 2015-2016 of the group SIGMORPHON (Cotterell et al., 2016). This inflectional morphology shared task aims to propose a system that solve reinflection problems based on the provided dataset.

The task actually contains three similar subtasks :

- **Inflection** : given a lemma and the target tag, the system should produce the right inflected form;

  English example

  Source lemma : do
  Target tag : Present participle
  Output : doing

- **Reinflection** : given a source tag and a source form (*i.e.* the source is no more a lemma) and the target tag, the system should produce the right inflected form;

  English example

  Source tag : Past
  Source form : did
  Target tag : Present participle
  Output : doing

- **Unlabeled Reinflection** : given only a source form (*i.e.* the system must recognize the morphosyntactic description of the source) and the target tag, the system should produce the right inflected form.

English example

Source form : did

Target tag : Present participle

Output : doing

There were several approaches proposed to solve these subtasks, they can be grouped into three types : pipelined approaches, neural approaches and approaches based on linguistic heuristics. None of them rely on analogy solving, which is the approach we want to explore. Recently (Murena et al., 2020) proposed an analogy based approach to tackle morphological tasks. We will discuss it in Section 3.3.

3.2 Dataset

The dataset released for this shared task contains training, development and test data as well as an evaluation script. Data is available for 10 languages : Spanish, German, Finnish, Russian, Turkish, Georgian, Navajo, Arabic, Hungarian and Maltese. Most of them are considered as languages with rich inflection.

All the provided files are in utf8 encoded text format. Each line of a file is an example for the task, the fields are separated by a tabulation. The forms and lemma are encoded as simple words while the tags are encoded as morphosyntactic descriptions (MSD).

Task 1 : Inflection

It consists in producing the right inflected form given a lemma and the target tag. The fields are thus LEMMA, MSD, TARGET FORM.

```
Forschung	pos=N,case=NOM,gen=FEM,num=PL	Forschungen
```

Listing 3.1: Example from the German development dataset for task 1

Task 2 : Reinflection

It consists in producing the right inflected form given given only a source form and the target tag. The fields are thus SOURCE MSD, SOURCE FORM, TARGET MSD, TARGET FORM.

```
pos=N,case=NOM,gen=FEM,num=SG	Forschung
pos=N,case=NOM,gen=FEM,num=PL	Forschungen
```

Listing 3.2: Example from the German development dataset for task 2

Task 3 : Unlabeled Reinflection

It consists in producing the right inflected form given a lemma and the target tag. The fields are thus SOURCE FORM, TARGET MSD, TARGET FORM.

```
Forschung	pos=N,case=NOM,gen=FEM,num=PL	Forschungen
```

Listing 3.3: Example from the German development dataset for task 3

Analogical equation : $van\ tu\ t : van\ t\ tu :: autopilot\ i\ t : x$

Result : $x = autopilot\ t\ i$

Transformation : (x, y, "t") $\longrightarrow$ x, "t", y

	vantut	autopilotit
x	van	autopilot
y	tu	i

Figure 3.1: Minimal complexity transformation for the analogy *vantut* : *vanttu* :: *autopilotit* : x

3.3 Kolmogorov Complexity

Several studies were conducted to try to solve morphological analogies (Hofstadter, 2002; Murena et al., 2020). In one article, they tried to use the semantic vector space model approach that was mentioned before in Section 2 to solve morphological analogies. But when they tried to adapt the parallelogram rule, it did not give the expected result due to the fact the algorithm word2vec can only work with words that are within the training data set (Murena et al., 2020). It was calculated by measuring the edit distance between sequences which is based on three edit operations between letters (Delhay & Miclet, 2004):

- inserting a letter in the target sequence;
- deleting a letter from the source sequence;
- replacing a letter in the source sequence with another letter in the target sequence.

The approach of (Murena et al., 2020) uses Kolmogorov Complexity to solve analogies by analyzing and calculating the distance between objects for proportional analogies. Their choice of analogies was limited to those that follow specific grammatical rules that are added to the main word forms (Murena et al., 2020). Those words are usually referred to as following the "regular word form changes." The focus was on the base word and its inflection. The association of transformation took both input (source term) and output (target term). Using a Python code, they built a set of analogical equations to represent the components of the source term and the additions needed to derive the target term.

The authors' assumption is the following: the rule explaining the shift from a word to its flexed form is unique and it is the one with the less complex description, *e.g.* the one with *minimal complexity transformations*.

They developed an algorithm to solve morphological analogies of the form $A : B :: C : X$ given A, B and C. The algorithm firstly builds a list of all the possible transformations such that applied, to a description[1] of A would produce B. Thus the algorithm must determine the transformation but also the description of the input words. To determine this

[1]A description of a word here is a tuple of letters and groups of letters such that the concatenation of the element of this tuple is the original word. For instance "vantut" can be described by ("vantut"), ("van", "tut"), ("van", "tu", "t"), *etc.*

description, all the possible morphological similarities between A and C are investigated. For each possibility, the transformation to apply to the description of A to produce B is determined. Eventually, the transformation considered as the right one is the one with minimal complexity transformations. FIGURE 3.1 describes the transformation with the minimal complexity for the analogical equation $vantut : vanttu :: autopilotit : x$.

Chapter 4

Conclusion and Future Work

This report gives a brief survey into the vast topic of analogies. As discussed, different approaches are used to tackle each type of analogy. Our focus for this project is on the semantic and morphological approaches. Though many articles have adopted different approaches to solve semantic analogies including Textual Analogy Parsing (TAP), Dependency Relations (DR), Vector Space Model (VSM) and learned Neural Network analogy, we also noticed that not many articles tried to tackle morphological analogies. Therefore, in this project, we want to work on analyzing morphological analogies by adapting a semantic approach. Instead of using the already existing approaches to analyze morphological analogies like Kolmogorov complexity and CopyCat, our idea is to make use of the semantic approaches and adapt them to solve morphological analogical equations between words.

In this project we are interested in morphological analogies, in particular, in working with inflectional affixes. We thus focus on reinflection as introduced in Subsection 3.2.2, following the shared task from 2015-2016 of the group SIGMORPHON (Cotterell et al., 2016). Due to the performance of the approach by (Lim et al., 2019), we intend to adapt the latter semantic approach to solve morphological analogies. This constitutes a novel contribution that differs from the current state of the art methods for solving morphological analogies. On the one hand, we wish to check whether we can transfer the semantic approach of (Lim et al., 2019) to solving morphological analogies. On the other hand, we hope to achieve competitive results to those of (Murena et al., 2020) with an alternative approach that is more straightforward than that based on minimum description length.

Bibliography

Banerjee, S., & Pedersen, T. (2003). Extended gloss overlaps as a measure of semantic relatedness. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, IJCAI'03, (p. 805–810). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Betrand (2016). Types of analogies. `https://magoosh.com/mat/types-of-analogies-on-the-mat/`.

Budanitsky, A., & Hirst, G. (2001). Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures. *Workshop on WordNet and Other Lexical Resources, Second meeting of the North American Chapter of the Association for Computational Linguistics, Pittsburgh, USA*, (pp. 29–34).

Chen, D., Peterson, J. C., & Griffiths, T. (2017). Evaluating vector-space models of analogy. In G. Gunzelmann, A. Howes, T. Tenbrink, & E. J. Davelaar (Eds.) *Proceedings of the 39th Annual Meeting of the Cognitive Science Society, CogSci 2017, London, UK, 16-29 July 2017*. cognitivesciencesociety.org.

Chiu, A., Poupart, P., & DiMarco, C. (2007). Generating lexical analogies using dependency relations. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, (pp. 561–570). Prague, Czech Republic: Association for Computational Linguistics.

Cotterell, R., Kirov, C., Sylak-Glassman, J., Yarowsky, D., Eisner, J., & Hulden, M. (2016). The sigmorphon 2016 shared task—morphological reinflection. In *Proceedings of the 2016 Meeting of SIGMORPHON*. Berlin, Germany: Association for Computational Linguistics.

Couceiro, M., Hug, N., Prade, H., & Richard, G. (2017). Analogy-preserving functions: A way to extend boolean samples. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, (pp. 1575–1581).

Daganzo, C. F. (1994). The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. *Transportation Research Part B: Methodological*, *28*(4), 269–287.

Deerwester, S., Dumais, S., Furnas, G., Landauer, T., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science 41*, (pp. 391–407).

Delhay, A., & Miclet, L. (2004). Analogical equations in sequences: Definition and resolution. In G. Paliouras, & Y. Sakakibara (Eds.) *Grammatical Inference: Algorithms and Applications*, (pp. 127–138). Berlin, Heidelberg: Springer Berlin Heidelberg.

Dingyi, L. (1985). Analogy and association. *Chinese Studies in Philosophy*, *16*(4), 92–107.

Do, Q. X., Lu, W., & Roth, D. (2012). Joint inference for event timeline construction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, (p. 677–687). USA: Association for Computational Linguistics.

Fibonicci (2019). Analogies - examples and types. `https://www.fibonicci.com/verbal-reasoning/analogies-examples/`.

Haspelmath, M. (2002). *Understanding morphology*. London: Arnold.
URL `https://doi.org/10.5281/zenodo.1236482`

Hofstadter, D. (2002). The copycat project: An experiment in nondeterminism and creative analogies.
URL `https://apps.dtic.mil/dtic/tr/fulltext/u2/a142744.pdf`

Hug, N., Prade, H., Richard, G., & Serrurier, M. (2016). Analogical classifiers: A theoretical perspective. In *ECAI 2016 - 22nd European Conference on Artificial Intelligence*, vol. 285 of *Frontiers in Artificial Intelligence and Applications*, (pp. 689–697). IOS Press.

Jiang, J. J., & Conrath, D. W. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of the 10th Research on Computational Linguistics International Conference*, (pp. 19–33). Taipei, Taiwan: The Association for Computational Linguistics and Chinese Language Processing (ACLCLP).

Kaveeta, V., & Lepage, Y. (2016). Solving analogical equations between strings of symbols using neural networks. In A. Coman, & S. Kapetanakis (Eds.) *Workshops Proceedings for the Twenty-fourth International Conference on Case-Based Reasoning (ICCBR 2016), Atlanta, Georgia, USA, October 31 - November 2, 2016*, vol. 1815 of *CEUR Workshop Proceedings*, (pp. 67–76). CEUR-WS.org.

Krott, A., Baayen, H., & Schreuder, R. (01 Mar. 2001). Analogy in morphology: modeling the choice of linking morphemes in dutch. *Linguistics*, *39*(1), 51 – 93.

Lafferty, J. D., McCallum, A., & Pereira, F. C. N. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, (p. 282–289). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Lamm, M., Chaganty, A., Jurafsky, D., Manning, C. D., & Liang, P. (2018a). Qsrl: A semantic role-labeling schema for quantitative facts. In M. El-Haj, P. Rayson, & A. Moore (Eds.) *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Paris, France: European Language Resources Association (ELRA).

Lamm, M., Chaganty, A. T., Manning, C. D., Jurafsky, D., & Liang, P. (2018b). Textual analogy parsing: What's shared and what's compared among analogous facts.

Landauer, T. K., & Dumais, S. T. (1997). A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, *104*, 211–240.

Lepage, Y. (2003). *De l'analogie rendant compte de la commutation en linguistique*. Habilitation 'a diriger des recherches, Universit'e Joseph-Fourier - Grenoble I.
URL `https://tel.archives-ouvertes.fr/tel-00004372`

Lesk, M. (1986). Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of the 5th Annual International Conference on Systems Documentation*, SIGDOC '86, (p. 24–26). New York, NY, USA: Association for Computing Machinery.

Lesk, M. E. (1969). Word-word associations in document retrieval systems. *American Documentation*, *20*(1), 27–38.

Levy, O., & Goldberg, Y. (2014). Dependency-based word embeddings. In *In: Proceedings of 52nd Annual Meeting of Association Computational Linguistics*, vol. 2: Short Papers, (p. 302–308).

Lim, S., Prade, H., & Richard, G. (2019). Solving word analogies: A machine learning perspective. In G. Kern-Isberner, & Z. Ognjanovic (Eds.) *Symbolic and Quantitative Approaches to Reasoning with Uncertainty, 15th European Conference, ECSQARU 2019, Belgrade, Serbia, September 18-20, 2019, Proceedings*, vol. 11726 of *Lecture Notes in Computer Science*, (pp. 238–250). Springer.

Lin, D. (1998). Automatic retrieval and clustering of similar words. In *Proceedings of the 17th international conference on Computational linguistics*, (pp. 768–774). Morristown, NJ, USA: Association for Computational Linguistics.

Mandy (2019). Machine learning & deep learning fundamentals.
URL `https://deeplizard.com/learn/video/YRhxdVk_sIs`

Medin, D. L., Goldstone, R. L., & Gentner, D. (1990). Similarity involving attributes and relations: Judgments of similarity and difference are not inverses. *Psychological Science*, *1*(1), 64–69.

Miclet, L., Bayoudh, S., & Delhay, A. (2008). Analogical dissimilarity: Definition, algorithms and two experiments in machine learning. *Journal of Artificial Intelligence Research*, *32*, 793–824.

Miclet, L., & Delhay, A. (2003). Analogy on Sequences : a Definition and an Algorithm. Research Report RR-4969, INRIA.

Murena, P.-A., Al-Ghossein, M., Dessalles, J.-L., & Cornuéjols, A. (2020). Solving analogies on words based on minimal complexity transformation. In C. Bessiere (Ed.) *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence,*

IJCAI-20, (pp. 1848–1854). International Joint Conferences on Artificial Intelligence Organization.

Pantel, P., & Lin, D. (2002). Discovering word senses from text. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '02, (pp. 613–619). New York, NY, USA: ACM.

Pennington, J., Socher, R., & Manning, C. (2014). GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (pp. 1532–1543). Doha, Qatar: Association for Computational Linguistics.

Prabhu (2018). Understanding of convolutional neural network (cnn) — deep learning. URL `https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148`

Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'95, (p. 448–453). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Roth, D., & Yih, W.-t. (2004). A linear programming formulation for global inference in natural language tasks. In *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004*, (pp. 1–8). Boston, Massachusetts, USA: Association for Computational Linguistics.

Ruge, G. (1992). Experiments on linguistically-based term associations. *Information Processing & Management*, *28*(3), 317 – 332.

Rumelhart, D. E., & Abrahamson, A. A. (1973). A model for analogical reasoning. *Cognitive Psychology*, *5*(1), 1 – 28.

Schiff, R., Bauminger, N., & Toledo, I. (2009). Analogical problem solving in children with verbal and nonverbal learning disabilities. *Journal of Learning Disabilities*, *42*(1), 3–13.

Turney, P. (2002). Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, (pp. 417–424). Philadelphia, Pennsylvania, USA: Association for Computational Linguistics.

Turney, P. D. (2001). Mining the web for synonyms: Pmi-ir versus lsa on toefl. In L. De Raedt, & P. Flach (Eds.) *Machine Learning: ECML 2001*, (pp. 491–502). Berlin, Heidelberg: Springer Berlin Heidelberg.

Turney, P. D. (2006). Similarity of semantic relations. *Comput. Linguist.*, *32*(3), 379–416.