Unsupervised Mining of Analogical Frames by Constraint Satisfaction

Lance De Vine **Shlomo Geva** **Peter Bruza**
Queensland University of Technology
Brisbane, Australia
{l.devine, s.geva, p.bruza}@qut.edu.au

Abstract

It has been demonstrated that vector-based representations of words trained on large text corpora encode linguistic regularities that may be exploited via the use of vector space arithmetic. This capability has been extensively explored and is generally measured via tasks which involve the automated completion of linguistic proportional analogies. The question remains, however, as to what extent it is possible to induce relations from word embeddings in a principled and systematic way, without the provision of exemplars or seed terms. In this paper we propose an extensible and efficient framework for inducing relations via the use of constraint satisfaction. The method is efficient, unsupervised and can be customized in various ways. We provide both quantitative and qualitative analysis of the results.

1 Introduction

The use and study of analogical inference and structure has a long history in linguistics, logic, cognitive psychology, scientific reasoning and education (Bartha, 2016), amongst others. The use of analogy has played an especially important role in the study of language, language change, and language acquisition and learning (Kiparsky, 1992). It has been a part of the study of phonology, morphology, orthography and syntactic grammar (Skousen, 1989), as well as the development of applications such as machine translation and paraphasing (Lepage and Denoual, 2005).

Recent progress with the construction of vector space representations of words based on their distributional profiles has revealed that analogical structure can be discovered and operationalised via the use of vector space algebra (Mikolov et al., 2013b). There remain many questions regarding the extent to which word vectors encode analogical structure and also the extent to which this structure can be uncovered. For example, we are not aware of any proposal or system that is focussed on the unsupervised and systematic discovery of analogies from word vectors that does not make use of exemplar relations, existing linguistic resources or seed terms. The automatic identification of linguistic analogies, however, offers many potential benefits for a diverse range of research and applications, including language learning and computational creativity.

Computational models of analogy have been studied since at least the 1960's (Hall, 1989; French, 2002) and have addressed tasks relating to both proportional and structural analogy. Many computational systems are built as part of investigations into how humans might perform analogical inference (Gentner and Forbus, 2011). Most make use of Structure Mapping Theory (SMT) (Gentner, 1983) or a variation thereof which maps one relational system to another, generally using a symbolic representation. Other systems use vector space representations constructed from corpora of natural language text (Turney, 2013). The analogies that are computed using word embeddings have primarily been proportional analogies and are closely associated with the prediction of relations between words. For example, a valid semantic proportional analogy is "cat is to feline as dog is to canine" which can be written as "cat : feline :: dog : canine."

In linguistics proportional analogies have been extensively studied in the context of both inflectional and derivational morphology (Blevins, 2016). Proportional analogies are used as part of an inference process to fill the cells/slots in a word *paradigm*. A paradigm is an array of morphological variations of a lexeme. For example, {cat, cats} is a simple singular-noun, plural-noun paradigm in English. Word paradigms exhibit inter-dependencies that facilitate the inference of

Lance De Vine, Shlomo Geva and Peter Bruza. 2018. Unsupervised Mining of Analogical Frames by Constraint Satisfaction. In *Proceedings of Australasian Language Technology Association Workshop*, pages 34−43.

new forms and for this reason have been studied within the context of language change. The informativeness of a form correlates with the degree to which knowledge of the form reduces uncertainty about other forms within the same paradigm (Blevins et al., 2017).

In this paper we propose a construction which we call an *analogical frame*. It is intended to elicit associations with the terms *semantic frame* and *proportional analogy*. It is an extension of a linguistic analogy in which the elements satisfy certain constraints that allow them to be induced in an unsupervised manner from natural language text. We expect that analogical frames will be useful for a variety of purposes relating to the automated induction of syntactic and semantic relations and categories .

The primary contributions of this paper are two-fold:

1. We introduce a generalization of proportional analogies with word embeddings which we call *analogical frames*.

2. We introduce an efficient constraint satisfaction based approach to inducing analogical frames from natural language embeddings in an unsupervised fashion.

In section 2 we present background and related research. In section 3 we present and explain the proposal of Analogical Frames. In section 4 we present methods implemented for ensuring search efficiency of Analogical Frames. In section 5 we present some analysis of empirical results. In section 6 we present discussion of the proposal and in section 7 we conclude.

2 Background and Related Work

2.1 Proportional Analogies

A proportional analogy is a 4-tuple which we write as $x_1 : x_2 :: x_3 : x_4$ and read as "x_1 is to x_2 as x_3 is to x_4", with the elements of the analogy belonging to some domain X (we use this notation as it is helpful later). From here-on we will use the term "analogy" to refer to proportional analogies unless indicated otherwise. Analogies can be defined over different types of domains, for example, strings, geometric figures, numbers, vector spaces, images etc. (Stroppa and Yvon, 2005) propose a definition of proportional analogy over any domain which is equipped with an internal composition law $\oplus$ making it a semi-group $(X, \oplus)$. This definition also applies to any richer algebraic structure such as groups or vector spaces. In $\mathbb{R}^n$, given x_1, x_2 and x_3 there is always only one point that can be assigned to x_4 such that proportionality holds. (Miclet et al., 2008) define a relaxed form of analogy which reads as "x_1 is to x_2 almost as x_3 is to x_4". To accompany this they introduce a measure of *analogical dissimilarity* (AD) which is a positive real value and takes the value 0 when the analogy holds perfectly. A set of four points $\mathbb{R}^n$ can therefore be scored for analogical dissimilarity and ranked.

2.2 Word Vectors and Proportional Analogies

The background just mentioned provides a useful context within which to place the work on linguistic regularities in word vectors (Mikolov et al., 2013b; Levy et al., 2014). (Mikolov et al., 2013b) showed that analogies can be completed using vector addition of word embeddings. This means that given x_1, x_2 and x_3 it is possible to infer the value of x_4. This is accomplished with the vector offset formula, or 3CosAdd (Levy et al., 2014).

$$\arg\max_{x_4} s(x_4, x_2 + x_3 - x_1) \qquad \text{3CosAdd}$$

The s in 3CosAdd is a similarity measure. In practice unit vectors are generally used with cosine similarity. (Levy et al., 2014) introduced an expression 3CosMul which tends to give a small improvement when evaluated on analogy completion tasks.

$$\arg\max_{x_4} \frac{s(x_4, x_3) \cdot s(x_4, x_2)}{s(x_4, x_1) + \epsilon} \qquad \text{3CosMul}$$

3CosAdd and 3CosMul are effectively scoring functions that are used to judge the correctness of a value for x_4 given values for x_1, x_2 and x_3.

2.3 Finding Analogies

Given that it is possible to complete analogies with word vectors it is natural to ask whether analogies can be identified without being given x_1, x_2 and x_3. (Stroppa and Yvon, 2005) considers an analogy to be valid when analogical proportions hold between all terms in the analogy. They describe a finite-state solver which searches for formal analogies in the domain of strings and trees. As noted by several authors (Lavallée and Langlais, 2009;

Langlais, 2016; Beltran et al., 2015) a brute force search to discovering proportional analogies is computationally difficult, with the complexity of a naive approach being at least $O(n^3)$ and perhaps $O(n^4)$. A computational procedure must at least traverse the space of all 3-tuples if assuming that the 4th term of an analogy can be efficiently inferred.

For example, if we use a brute force approach to discovering linguistic analogies using a vocabulary of 100 000 words, we would need to examine all combinations of 4-tuples, or 100000^4, or 10^{20} combinations. Various strategies may be

$$\begin{tikzcd} \boldsymbol{x_1} \arrow[r, dash] \arrow[d, dash] & \boldsymbol{x_2} \arrow[d, dash] \\ \boldsymbol{x_3} \arrow[r, dash] & \boldsymbol{x_4} \end{tikzcd}$$

Figure 1: A Proportional Analogy template, with 4 variables to be assigned appropriate values

considered for making this problem tractable. The first observation is that symmetries of an analogy should not be recomputed (Lepage, 2014). This can reduce the compute time by 8 for a single analogy as there are 8 symmetric configurations.

Another proposed strategy is the construction of a feature tree for the rapid computation and analysis of tuple differences over vectors with binary attributes (Lepage, 2014). This method was used to discover analogies between images of Chinese characters. This has complexity $O(n^2)$. It computes the $\frac{n(n+1)}{2}$ vectors between pairs of tuples and collects them together into clusters containing the same difference vector. A variation of this method was reported in (Fam and Lepage, 2016) and (Fam and Lepage, 2017) for automatically discovering *analogical grids* of word paradigms using edit distances between word strings. It is not immediately obvious, however, how to extend this to the case of word embeddings where differences between word representations are real valued vectors.

A related method is used in (Beltran et al., 2015) for identifying analogies in relational databases. It is less constrained as it uses analogical dissimilarity as a metric when determining valid analogies.

(Langlais, 2016) extend methods from (Lepage, 2014) to scale to larger datasets for the purpose of machine translation, but also limit themselves to the formal or graphemic level instead of more general semantic relations between words.

2.4 Other Related Work

The present work is related to a number of research themes in language learning, relational learning and natural language processing. We provide a small sample of these.

(Holyoak and Thagard, 1989) introduce the use of constraint satisfaction as a key requirement for models of analogical mapping. Their computer program ACME (Analogical Constraint Mapping Engine) uses a connectionist network to balance structural, semantic and pragmatic constraints for mapping relations. (Hummel and Holyoak, 1997) propose a computational model of analogical inference and schema induction using distributed patterns for representing objects and predicates. (Doumas et al., 2008) propose a computational model which provides an account of how structured relation representations can be learned from unstructured data. More specifically to language acquisition, (Bod, 2009) uses analogies over trees to derive and analyse new sentences by combining fragments of previously seen sentences. The proposed framework is able to replicate a range of phenomena in language acquisition.

From a more cognitive perspective (Kurtz et al., 2001) investigates how mutual alignment of two situations can create better understanding of both. Related to this (Gentner, 2010), and many others, argue that analogical ability is the key factor in human cognitive development.

(Turney, 2006, 2013) makes extensive investigations of the use of corpus based methods for determining relational similarities and predicting analogies.

(Miclet and Nicolas, 2015) propose the concept of an analogical complex which is a blend of analogical proportions and formal concept analysis.

More specifically in relation to word embeddings, (Zhang et al., 2016) presents an unsupervised approach for explaining the meaning of terms via word vector comparison.

In the next section we describe an approach which addresses the task of inducing analogies in an unsupervised fashion from word vectors and builds on existing work relating to word embeddings and linguistic regularities.

3 Analogical Frames

The primary task that we address in this paper is the discovery of linguistic proportional analogies in an unsupervised fashion given only the distributional profile of words. The approach that we take is to consider the problem as a constraint satisfaction problem (CSP) (Rossi et al., 2006). We increase the strength of constraints until we can accurately decide when a given set of words and their embeddings forms a valid proportional analogy. At this point we introduce some terminology:

Constraint satisfaction problems are generally defined as a triple $P = \langle X, D, C \rangle$ where
$X = \{x_1, ..., x_n\}$ is a set of variables.
$D = \{d_1, ..., d_n\}$ is the set of domains associated with the variables.
$C = \{c_1, ..., c_m\}$ is the set of constraints to be satisfied.

A solution to a constraint satisfaction problem must assign a single value to each variable $x_1, ..., x_n$ in the problem. There may be multiple solutions. In our problem formulation there is only one domain which is the set of word types which comprise the vocabulary. Each variable must be assigned a word identifier and each word is associated with one or more vector space representations. Constraints on the words and associated vector space representation limit the values that the variables can take. From here-on we will we use the bolded symbol $\boldsymbol{x_i}$ to indicate the vector space value of the word assigned to the variable x_i.

In our proposal we use the following five constraints.

C1. AllDiff constraint. The *Alldiff* constraint constrains all terms of the analogy to be distinct (The Alldiff constraint is a common constraint in CSPs), such that $x_i \neq x_j$ for all $1 < i < j < n$.

C2. Asymmetry constraint. The *asymmetry constraint* (Meseguer and Torras, 2001) is used to eliminate unnecessary searches in the search tree. It is defined as a partial ordering on the values of a subset of the variables. In the case of a 2x3 analogical frame (figure 2), for example, we define the ordering as:

$$x_1 \prec x_2 \prec x_3, \text{ and } x_1 \prec x_4.$$

where the ordering is defined on the integer identifiers of the words in the vocabulary.

C3. Neighbourhood Constraint. The *neighbourhood constraint* is used to constrain the value of variables to the words which are within the nearest neighbourhood of the words to which they are connected. We define this as:

$$x_i \in Neigh_t(x_j) \text{ and } x_j \in Neigh_t(x_i)$$

where $Neigh_t(x_i)$ is the nearest neighbourhood of t words of the the word assigned to x_i, as measured in the vector space representation of the words.

C4. Parallel Constraint. The *Parallel Constraint* forces opposing difference vectors to have a minimal degree of parallelism.

$$\widehat{\boldsymbol{x_2} - \boldsymbol{x_1}} \cdot \widehat{\boldsymbol{x_5} - \boldsymbol{x_4}} < pThreshold$$

where pThreshold is a parameter. For the parallel constraint we ensure that the difference vector $\boldsymbol{x_2} - \boldsymbol{x_1}$ has a minimal cosine similarity to the difference vector $\boldsymbol{x_5} - \boldsymbol{x_1}$. This constraint overlaps to some extent with the proportionality constraint (below), however it serves a different purpose, which is to eliminate low probability candidate analogies.

C5. Proportionality Constraint. The *Proportionality Constraint* constrains the vector space representation of words to form approximate geometric proportional analogies. It is a quarternary constraint. For any given 4-tuple, we use the concept of "inter-predictability" (Blevins et al., 2017) to decide whether the 4-tuple is acceptable. We enforce inter-predictability by requiring that each term in a 4-tuple is predicted by the other three terms. This implies four analogy completion tasks which must be satisfied for the variable assignment to be accepted.

$$\boldsymbol{x_1} : \boldsymbol{x_2} :: \boldsymbol{x_3} : x \Rightarrow x = \boldsymbol{x_4}$$
$$\boldsymbol{x_2} : \boldsymbol{x_1} :: \boldsymbol{x_4} : x \Rightarrow x = \boldsymbol{x_3}$$
$$\boldsymbol{x_3} : \boldsymbol{x_4} :: \boldsymbol{x_1} : x \Rightarrow x = \boldsymbol{x_2}$$
$$\boldsymbol{x_4} : \boldsymbol{x_3} :: \boldsymbol{x_2} : x \Rightarrow x = \boldsymbol{x_1}$$

(Stroppa and Yvon, 2005) use a similar approach with exact formal analogies. With our approach, however, we complete analogies using word vectors and analogy completion formulas (eg. using 3CosAdd or 3CosMul, or a derivative).

The proportionality constraint is a relatively expensive constraint to enforce as it requires many vector-vector operations and comparison against all word vectors in the vocabulary. This constraint may be checked approximately which we discuss in the next section.

3.1 The Insufficiency of 2x2 Analogies

When we experiment with discovering 2x2 analogies using the constraints just described we find that we can't easily set the parameters of the constraints such that only valid analogies are produced, without severely limiting the types of analogies which we accept. For example, the following analogy is produced "oldest : old :: earliest : earlier". We find that these types of mistakes are common. We therefore take the step of expanding the number of variables so that the state space is now larger (figure 2).

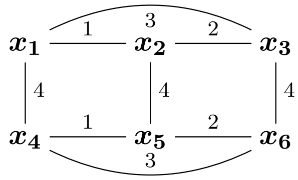


Figure 2: A 2x3 Analogical Frame

The idea is to increase the inductive support for each discovered analogy by requiring that analogies be part of a larger system of analogies. We refer to the larger system of analogies as an *Analogical Frame*. It is important to note that in figure 2, x_1 and x_3 are connected. The numbers associated with the edges indicate aligned vector differences. It is intended that the analogical proportions hold according to the connectivity shown. For example, proportionality should hold such that $\boldsymbol{x_1} : \boldsymbol{x_2} :: \boldsymbol{x_4} : \boldsymbol{x_5}$ and $\boldsymbol{x_1} : \boldsymbol{x_3} :: \boldsymbol{x_4} : \boldsymbol{x_6}$ and $\boldsymbol{x_2} : \boldsymbol{x_3} :: \boldsymbol{x_5} : \boldsymbol{x_6}$. It is also important to note that while we have added only two new variables, we have increased the number of constraints by almost 3 times. It is not exactly 3 because there is some redundancy in the proportions.

3.2 New Formulas

We now define a modified formula for completing analogies that are part of a larger systems of analogies such as the analogical frame in figure 2. It can be observed that 3CosAdd and 3CosMul effectively assigns a score to vocabulary items and then selects the item with the largest score. We do the same but with a modified scoring function which is a hybrid of 3CosAdd and 3CosMul. 3CosAdd or 3CosMul could both be used as part of our approach, but the formula which we propose, better captures the intuition of larger systems of analogies where the importance is placed on 1) symmetry, and 2) average offset vectors. When we are not given any prior knowledge, or exemplar, there is no privileged direction within an analogical frame. For example, in figure 2, the pair (x_2, x_5) has the same importance as (x_4, x_5).

We first construct difference vectors and then average those that are aligned.

$$dif_{2,1} = \widehat{\boldsymbol{x_2} - \boldsymbol{x_1}} \qquad dif_{4,1} = \widehat{\boldsymbol{x_4} - \boldsymbol{x_1}}$$
$$dif_{5,4} = \widehat{\boldsymbol{x_5} - \boldsymbol{x_4}} \qquad dif_{5,2} = \widehat{\boldsymbol{x_5} - \boldsymbol{x_2}}$$
$$dif_{6,3} = \widehat{\boldsymbol{x_6} - \boldsymbol{x_3}}$$
$$difsum_1 = dif_{2,1} + dif_{5,4}$$
$$difsum_2 = dif_{4,1} + dif_{5,2} + dif_{6,3}$$
$$\boldsymbol{dif_1} = \frac{difsum_1}{|difsum_1|}$$
$$\boldsymbol{dif_2} = \frac{difsum_2}{|difsum_2|}$$

The vector $\boldsymbol{dif_1}$ is the normalized average offset vector indicated with a 1 in figure 2. The vector $\boldsymbol{dif_2}$ is the normalized average offset vector indicated with a 4 in figure 2.

Using these normalized average difference vectors we define the scoring function for selecting x_5 given x_1, x_2 and x_4 as:

$$\arg\max_{x_5} \; s(\boldsymbol{x_5}, \boldsymbol{x_4}) \cdot s(\boldsymbol{x_5}, \boldsymbol{dif_1}) + s(\boldsymbol{x_5}, \boldsymbol{x_2}) \cdot s(\boldsymbol{x_5}, \boldsymbol{dif_2}) \tag{1}$$

The formulation makes use of all information available in the analogical frame. Previous work has provided much evidence for the linear compositionality of word vectors as embodied by 3CosAdd (Vylomova et al., 2015; Hakami et al., 2017). It has also been known since (Mikolov et al., 2013a) that averaging the difference vectors of pairs exhibiting the same relation results in a difference vector with better predictive power for that relation (Drozd et al., 2016).

Extrapolating from figure 2 larger analogical frames can be constructed, such as 2 x 4, 3 x 3, or 2 x 2 x 3. Each appropriately connected 4-tuple contained within the frame should satisfy the proportionality constraint.

4 Frame Discovery and Search Efficiency

The primary challenge in this proposal is to efficiently search the space of variable assignments. We use a depth first approach to cover the search space as well several other strategies to make this search efficient.

4.1 Word Embedding Neighbourhoods

The most important strategy is the concentration of compute resources on the nearest neighbourhoods of word embeddings. Most analogies involve terms that are within the nearest neighborhood of each other when ranked according to similarity (Linzen, 2016). We therefore compute the nearest neighbour graph of every term in the vocabulary as a pre-processing step and store the result as an adjacency list for each vocabulary item. We do this efficiently by using binary representations of the word embeddings (Jurgovsky et al., 2016) and re-ranking using the full precision word vectors. The nearest neighbour list of each vocabulary entry is used in two different ways, 1) for exploring the search space of variable assignments, and 2) efficiently eliminating variable assignments (next section) that do not satisfy the proportional analogy constraints.

When traversing the search tree we use the adjacency list of an already assigned variable to assign a value to a nearby variable in the frame. The breadth of the search tree at each node is therefore parameterized by a global parameter t assigned by the user. The parameter t is one of the primary parameters of the algorithm and will determine the length of the search and the size of the set of discovered frames.

4.2 Elimination of Candidates by Sampling

A general principle used in most CSP solving is the quick elimination of improbable solutions. When we assign values to variables in a frame we need to make sure that the proportional analogy constraint is satisfied. For four given variable assignments w_1, w_2, w_3 and w_4 this involves making sure that w_4 is the best candidate for completing the proportional analogy involving w_1, w_2 and w_3. This is equivalent to knowing if there are any better vocabulary entries than w_4 for completing the analogy. Instead of checking all vocabulary entries we can limit the list of entries to the m nearest neighbours of w_2, w_3 and w_4 [1] to check if any score higher than w_4. If any of them score higher the proportional analogy constraint is violated and the variable assignment is discarded. The advantage of this is that we only need to check 3 x m entries to approximately test the correctness of w_4. We find that if we set m to 10 then most incorrect analogies are eliminated. An exhaustive check for correctness can be made as a post processing step.

[1] assuming w_1 is farthest away from w_4 and that the nearest neighbours of w_1 are not likely to help check the correctness of w_4

4.3 Other Methods

Another important method for increasing efficiency is testing the degree to which opposing difference vectors in a candidate proportional analogy are parallel to each other. This is encapsulated in constraint C4. While using parallelism as a parameter for solving word analogy problems has not been successful (Levy et al., 2014), we have found that the degree of parallelism is a good indicator of the confidence that we can have in the analogy once other constraints have been satisfied. Other methods employed to improve efficiency include the use of Bloom filters to test neighbourhood membership and the indexing of discovered proportions so as not to repeat searching.

4.4 Extending Frames

Frames can be extended by extending the initial *base frame* in one or more directions and searching for additional variable assignments that satisfy all constraints. For example, a 2x3 frame can be extended to a 2x4 frame by assigning values to two new variables x_7 and x_8 (figure 3).

$$\begin{tikzcd}
\boldsymbol{x_1} \arrow[r, dash] \arrow[d, dash] & \boldsymbol{x_2} \arrow[r, dash] \arrow[d, dash] & \boldsymbol{x_3} \arrow[r, dashed] \arrow[d, dash] & \boldsymbol{x_7} \\
\boldsymbol{x_4} \arrow[r, dash] & \boldsymbol{x_5} \arrow[r, dash] & \boldsymbol{x_6} \arrow[r, dashed] & \boldsymbol{x_8}
\end{tikzcd}$$

Figure 3: Extending a 2x3 frame

As frames are extended the average offset vectors (equation 1) are recomputed so that the offset vectors become better predictors for computing proportional analogies.

5 Experimental Setup and Results

The two criteria we use to measure our proposal include a) accuracy of analogy discovery, b) compute scalability. Other criteria are also possible such as relation diversity and interestingness of relations.

The primary algorithm parameters are 1) The number of terms in the vocabulary to search, 2) the size of the nearest neighbourhood of each term to search, 3) the degree of parallelism required for opposing vector differences and 4) whether to extend base frames.

5.1 Retrieving Analogical Completions from Frames

When frames are discovered they are stored in a *Frame Store*, a data structure used to efficiently store and retrieve frames. Frames are indexed using posting lists similar to a document index. To complete an incomplete analogy, all frames which contain all terms in the incomplete analogy are retrieved. For each retrieved frame the candidate term for completing the analogy is determined by cross-referencing the indices of the terms within the frame (figure 4). If diverse candidate terms are selected from multiple frames, voting is used to select the final analogy completion, or random selection in the case of tied counts.

$$a_{1,1} : a_{1,3} :: a_{3,1} :?$$

$$\begin{array}{ccc} \mathbf{a_{1,1}} & a_{1,2} & \mathbf{a_{1,3}} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ \mathbf{a_{3,1}} & a_{3,2} & \underline{a_{3,3}} \end{array}$$

Figure 4: Determining an analogy completion from a larger frame

We conducted experiments with embeddings constructed by ourselves as well as with publicly accessible embeddings from the fastText web site[2] trained on 600B tokens of the Common Crawl (Mikolov et al., 2018) .

We evaluated the accuracy of the frames by attempting to complete the analogies from the well known Google analogy test set.[3] The greatest challenge in this type of evaluation is adequately covering the evaluation items. At least three of the terms in an analogy completion item need to be simultaneously present in a single frame for the item to be attempted. We report the results of a typical execution of the system using a nearest neighbourhood size of 25, a maximum vocabulary size of 50 000, and a minimal cosine similarity between opposing vector differences of 0.3. For this set of parameters, 8589 evaluation items were answered by retrieval from the frame store, covering approximately 44% of the evaluation items (table 1). Approximately 30% of these were from the semantic category, and 70% from the syntactic. We compared the accuracy of completing analogies using the frame store, to the accuracy of both 3CosAdd and 3CosMul using the same embeddings as used to build the frames.

Table 1: Analogy Completion on Google Subset

	3CosAdd	3CosMul	Frames
Sem. (2681)	2666	2660	**2673**
Syn. (5908)	5565	5602	**5655**
Tot. (8589)	8231	8262	**8328**

Results show that the frames are slightly more accurate than 3CosAdd and 3CosMul, achieving 96.9% on the 8589 evaluation items. It needs to be stressed, however, that the objective is not to outperform vector arithmetic based methods, but rather to verify that the frames have a high degree of accuracy.

To better determine the accuracy of the discovered frames we also randomly sampled 1000 of the 21571 frames generated for the results shown in table 2, and manually checked them. The raw outputs are included in the online repository[4]. These frames cover many relations not included in the Google analogy test set. We found 9 frames with errors giving an accuracy of 99.1%.

It should be noted that the accuracy of frames is influenced by the quality of the embeddings. However, even with embeddings trained on small corpora it is possible to discover analogies provided that sufficient word embedding training epochs have been completed.

The online repository contains further empirical evaluations and explanations regarding parameter choices, including raw outputs and errors made by the system.

5.2 Scaling: Effect of Neighbourhood Size and pThreshold

Tables 2 and 3 show the number of frames discovered on typical executions of the software. The reported numbers are intended to give an indication of the relationship between neighbourhood size, number of frames produced and execution time. The reported times are for 8 software threads.

5.3 Qualitative Analysis

From inspection of the frames we see that a large part of the relations discovered are grammatical or morpho-syntactic, or are related to high frequency

[2]https://fasttext.cc/docs/en/english-vectors.html

[3]http://download.tensorflow.org/data/questions-words.txt

[4]https://github.com/ldevine/AFM

Table 2: Par Thres = 0.3 (Less Constrained)

Near. Sz.	15	20	25	30
Frames	13282	16785	19603	21571
Time (sec)	20.1	29.3	38.3	45.9

Table 3: Par Thres = 0.5 (More Constrained)

Near. Sz.	15	20	25	30
Frames	6344	7995	9286	10188
Time (sec)	10.4	15.7	21.1	26.0

entities. However we also observe a large number of other types of relations such as synonyms, antonyms, domain alignments and various syntactic mappings. We provide but a small sample below.

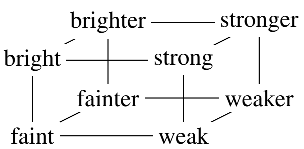
eyes	eye	blindness
ears	ear	deafness

Father	Son	Himself
Mother	Daughter	Herself

geschichte	gesellschaft	musik
histoire	societe	musique

aircraft	flew	skies	air
ships	sailed	seas	naval

always	constantly	constant
often	frequently	frequent
sometimes	occasionally	occasional



(2 x 2 x 2 Frame)

We have also observed reliable mappings with embeddings trained on non-English corpora.

The geometry of analogical frames can also be explored via visualizations of projections onto 2D sub-spaces derived from the offset vectors. [5]

6 Discussion

We believe that the constraint satisfaction approach introduced in this paper is advantageous because it is a systematic but flexible and can make use of methods from the constraint satisfaction domain. We have only mentioned a few of the primary CSP concepts in this paper. Other constraints can be included in the formulation such as set membership constraints where sets may be clusters, or documents.

One improvement that could be made to the proposed system is to facilitate the discovery of relations that are not one-to-one. While we found many isolated examples of one-to-many relations expressed in the frames, a strictly symmetrical proportional analogy does not seem ideal for capturing one-to-many relations.

As outlined by (Turney, 2006) there are many applications of automating the construction and/or discovery of analogical relations. Some of these include relation classification, metaphor detection, word sense disambiguation, information extraction, question answering and thesaurus generation.

Analogical frames should also provide insight into the geometry of word embeddings and may provide an interesting way to measure their quality.

The most interesting application of the system is in the area of computational creativity with a human in the loop. For example, analogical frames could be chosen for their interestingness and then expanded.

6.1 Software and Online Repository

The software implementing the proposed system as a set of command line applications can be found in the online repository[6]. The software is implemented in portable C++11 and compiles on both Windows and Unix based systems without compiled dependencies. Example outputs of the system as well as parameter settings are provided in the online repository including the outputs created from embeddings trained on a range of corpora.

7 Future Work and Conclusions

Further empirical evaluation is required. The establishment of more suitable empirical benchmarks for assessing the effectiveness of open analogy discovery is important. The most interesting potential application of this work is in the combination of automated discovery of analogies and human judgment. There is also the possibility of establishing a more open-ended compute architecture that could search continuously for analogical frames in an online fashion.

[5] Examples provided in the online repository

[6] https://github.com/ldevine/AFM

References

Paul Bartha. 2016. Analogy and analogical reasoning. *The Stanford Encyclopedia of Philosophy*.

William Correa Beltran, Hélène Jaudoin, and Olivier Pivert. 2015. A clustering-based approach to the mining of analogical proportions. In *Tools with Artificial Intelligence (ICTAI), 2015 IEEE 27th International Conference on*, pages 125–131. IEEE.

James P Blevins. 2016. *Word and paradigm morphology*. Oxford University Press.

James P Blevins, Farrell Ackerman, Robert Malouf, J Audring, and F Masini. 2017. Word and paradigm morphology.

Rens Bod. 2009. From exemplar to grammar: A probabilistic analogy-based model of language learning. *Cognitive Science*, 33(5):752–793.

Leonidas AA Doumas, John E Hummel, and Catherine M Sandhofer. 2008. A theory of the discovery and predication of relational concepts. *Psychological review*, 115(1):1.

Aleksandr Drozd, Anna Gladkova, and Satoshi Matsuoka. 2016. Word embeddings, analogies, and machine learning: Beyond king-man+ woman= queen. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3519–3530.

Rashel Fam and Yves Lepage. 2016. Morphological predictability of unseen words using computational analogy.

Rashel Fam and Yves Lepage. 2017. A study of the saturation of analogical grids agnostically extracted from texts.

Robert M French. 2002. The computational modeling of analogy-making. *Trends in cognitive Sciences*, 6(5):200–205.

Dedre Gentner. 1983. Structure-mapping: A theoretical framework for analogy. *Cognitive science*, 7(2):155–170.

Dedre Gentner. 2010. Bootstrapping the mind: Analogical processes and symbol systems. *Cognitive Science*, 34(5):752–775.

Dedre Gentner and Kenneth D Forbus. 2011. Computational models of analogy. *Wiley interdisciplinary reviews: cognitive science*, 2(3):266–276.

Huda Hakami, Kohei Hayashi, and Danushka Bollegala. 2017. An optimality proof for the pairdiff operator for representing relations between words. *arXiv preprint arXiv:1709.06673*.

Rogers P Hall. 1989. Computational approaches to analogical reasoning: A comparative analysis. *Artificial intelligence*, 39(1):39–120.

Keith J Holyoak and Paul Thagard. 1989. Analogical mapping by constraint satisfaction. *Cognitive science*, 13(3):295–355.

John E Hummel and Keith J Holyoak. 1997. Distributed representations of structure: A theory of analogical access and mapping. *Psychological review*, 104(3):427.

Johannes Jurgovsky, Michael Granitzer, and Christin Seifert. 2016. Evaluating memory efficiency and robustness of word embeddings. In *European Conference on Information Retrieval*, pages 200–211. Springer.

Paul Kiparsky. 1992. Analogy. *International Encyclopedia of Linguistics*.

Kenneth J Kurtz, Chun-Hui Miao, and Dedre Gentner. 2001. Learning by analogical bootstrapping. *The Journal of the Learning Sciences*, 10(4):417–446.

Philippe Langlais. 2016. Efficient identification of formal analogies.

Jean-François Lavallée and Philippe Langlais. 2009. Unsupervised morphological analysis by formal analogy. In *Workshop of the Cross-Language Evaluation Forum for European Languages*, pages 617–624. Springer.

Yves Lepage. 2014. Analogies between binary images: Application to chinese characters. In *Computational Approaches to Analogical Reasoning: Current Trends*, pages 25–57. Springer.

Yves Lepage and Etienne Denoual. 2005. Purest ever example-based machine translation: Detailed presentation and assessment. *Machine Translation*, 19(3-4):251–282.

Omer Levy, Yoav Goldberg, and Israel Ramat-Gan. 2014. Linguistic regularities in sparse and explicit word representations. In *CoNLL*, pages 171–180.

Tal Linzen. 2016. Issues in evaluating semantic spaces using word analogies. *arXiv preprint arXiv:1606.07736*.

Pedro Meseguer and Carme Torras. 2001. Exploiting symmetries within constraint satisfaction search. *Artificial intelligence*, 129(1-2):133–163.

Laurent Miclet, Sabri Bayoudh, and Arnaud Delhay. 2008. Analogical dissimilarity: definition, algorithms and two experiments in machine learning. *Journal of Artificial Intelligence Research*, 32:793–824.

Laurent Miclet and Jacques Nicolas. 2015. From formal concepts to analogical complexes. In *CLA 2015*, page 12.

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

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013b. Linguistic regularities in continuous space word representations. In *HLT-NAACL*, volume 13, pages 746–751.

Francesca Rossi, Peter Van Beek, and Toby Walsh. 2006. *Handbook of constraint programming*. Elsevier.

Royal Skousen. 1989. *Analogical modeling of language*. Springer Science & Business Media.

Nicolas Stroppa and Francçois Yvon. 2005. An analogical learner for morphological analysis. In *Proceedings of the Ninth Conference on Computational Natural Language Learning*, pages 120–127. Association for Computational Linguistics.

Peter D Turney. 2006. Similarity of semantic relations. *Computational Linguistics*, 32(3):379–416.

Peter D Turney. 2013. Distributional semantics beyond words: Supervised learning of analogy and paraphrase. *arXiv preprint arXiv:1310.5042*.

Ekaterina Vylomova, Laura Rimell, Trevor Cohn, and Timothy Baldwin. 2015. Take and took, gaggle and goose, book and read: Evaluating the utility of vector differences for lexical relation learning. *arXiv preprint arXiv:1509.01692*.

Yating Zhang, Adam Jatowt, and Katsumi Tanaka. 2016. Towards understanding word embeddings: Automatically explaining similarity of terms. In *Big Data (Big Data), 2016 IEEE International Conference on*, pages 823–832. IEEE.