IR meets NLP: On the Semantic Similarity between Subject-Verb-Object Phrases

Dmitrijs Milajevs, Mehrnoosh Sadrzadeh and Thomas Roelleke
School of Electronic Engineering and Computer Science, Queen Mary University of London
London, UK
d.milajevs@qmul.ac.uk, m.sadrzadeh@qmul.ac.uk, t.roelleke@qmul.ac.uk

ABSTRACT

Measuring the semantic similarity between phrases and sentences is an important task in natural language processing (NLP) and information retrieval (IR). We compare the quality of the distributional semantic NLP models against phrase-based semantic IR. The evaluation is based on the correlation between human judgements and model scores on a distributional phrase similarity task. We experiment with four NLP and two IR model variants. On the NLP side, models vary over normalization schemes and composition operators. On the IR side, models vary with respect to estimation of the probability of a term being in a document, namely $P(t|d)$ where only term co-occurrence information is used and $P(t|d, \text{sim})$ which incorporates term distributional similarity. A mixture of the two methods is presented and evaluated. For both methods, word meanings are derived from large corpora of data: the BNC and ukWaC. One of the main findings is that grammatical distributional models give better scores than the IR models. This suggests that an IR model enriched with distributional linguistic information performs better in the long standing problem in IR of document retrieval where there is no direct symbolic relationship between query and document concepts.

Categories and Subject Descriptors

I.2.7 [**Artificial Intelligence**]: Natural Language Processing; H.3.3 [**Information Search and Retrieval**]: Retrieval Models

General Terms

Algorithms; Theory

Keywords

NLP; Distributional Semantics; Semantic IR; Subject-Verb-Object (SVO) Phrases.

1. INTRODUCTION

Traditional information retrieval (IR) deals with document retrieval where a similarity measure is applied to rank documents with respect to a query, and the measure is usually based on *word* frequencies (within-document frequencies and collection-wide frequencies) and involves frequency and length normalisations. Natural language processing (NLP) is concerned about preserving and utilising the *meaning* of words and sentences where words representations are composed using grammatical relations to form sentence representations. We compare in this paper the methods of semantic phrase-based IR and distributional vector-based NLP and investigate the integration of their techniques. We focus on "subject verb object" phrases (in NLP terminology: simple transitive sentences) and measure and utilise their semantic similarities

The first challenge is finding an appropriate answer to the question: what is the semantic similarity between

> "agent sells property" and "family buys home"?

These are two subject-verb-object phrases and they contain different words, with regard to traditional IR, there is a "term mismatch" problem. Even though there is no syntactic word match, the phrases are semantically related. In this case, this is because if an agent sells a property, then this potentially implies that a family buys a home. Also, from an ontology point of view, a "home" is a "property", and this makes the objects to be related. Finally, despite the fact that there is no obvious relationship between "agent" and "family", the verbs "buys" and "sells" are strongly semantically related. Consequently, one has to provide an answer to the sequel question: how one can quantify similarity between two phrases?

To illustrate different levels of similarity, consider two phrases that deem to be semantically more similar than the phrases above:

> "woman drinks water" and "wife pours tea".

The phrases are more related than the phrases of the first example, because there is also a strong semantic relationship between the subjects, which is not present in the first example.

The research questions motivating this work are: how do NLP models compare to IR models? Under which conditions each approach works best? And how can one combine them? We experiment with a variety of semantic phrase-based IR and distributional vector-based NLP models and address these.

1.1 Structure and Contributions

The remainder of this paper is structured as follows. Section 2 provides the background on the distributional vector-based NLP models and their compositionality, IR models and phrase-based semantic IR. Section 3 explains the task and methodologies for solving it. Section 4 compares the DS and IR models and presents an integrated model. Section 5 goes over the evaluation methodology and experiment set up. Section 6 presents results and offers an analysis. Finally, Section 7 concludes the work.

ICTIR'15, September 27–30, Northampton, MA, USA.

DOI: http://dx.doi.org/10.1145/2808194.2809448.

The main contributions of this paper are the comparison of DS-based and IR-based techniques. Another contribution is the investigation of the performance of an integrated DS+IR model.

2. BACKGROUND

2.1 Vector-based Distributional NLP

NLP models reason about meanings of words and phrases of language (we follow the IR terminology and throughout the paper use "phrase" to refer to a sequence of words, including sentences). They can be classified into two broad categories: statistical and rule-based. The former represents meanings of words based on the likelihood of their occurrence in text; examples herein are Markov-based models, distributional semantics, and neural embeddings. The latter offer a logical analysis of the structures of phrases and sentences based on the grammatical rules of language. Examples thereof are Montague semantics, type-logical grammars, and context-free grammars. For the purpose of this paper, we work with distributional semantics shorthanded to DS (see [33] for an overview). These have been shown to lend themselves to both word and phrase level analysis and constitute an active field of research.

Distributional semantics is based on the idea of Firth [10] that the meaning of a word depends on the contexts in which it often occurs. Hence, words that often occur close to same features have similar meanings. For example, 'boat' and 'ship' have similar meanings, since they both occur close to 'ocean' and 'sail'. Their meaning is not similar to that of 'cat' and 'dog', since the latter often occur close to 'pet' and 'furry'.

These models are formalized in the form of a vector space whose basis vectors are a fixed set of features [23]. The meaning of a words of interest, referred to as a *target* word, is a vector in this vector space. Each such vector is built after fixing a corpus of text (a large collection of documents), a set of features, and a neighboring window of k words (e.g. 5). For a target word t and a feature f, raw frequencies are counted by summing the number of times t occurred k-words on either side of f, divided by the number of times t got counted (i.e. the size of the window). Dividing this by the total number of times t occurred in the corpus (e.g. L), provides a probabilistic measure of the raw counts:

$$\text{freq}_\text{k}(t) = \frac{\sum_f N(f,t)}{k} \qquad P_k(t) = \frac{\text{freq}_\text{k}(t)}{L}$$

Distributional models have been applied to different language tasks, for example, in [31] they were applied to reason about word synonymy, in [21] they were used for induction and knowledge acquisition, in [32] and [22] they were used for modelling word sense discrimination and clustering. Many of these applications rely on the presence of quantitative measures in the vector space to represent the similarity in meaning using the distance between the vectors. In a study by Bullinaria and Levy [4], it was shown that the geometric distance based on the cosine of the angle between the vectors performed best in semantic word similarity tasks $\text{sim}(w, w') = \cos(\overrightarrow{w}, \overrightarrow{w}')$. In this paper we focus on the cosine and do not extend on other measures such as Jaccard, Dice, and Tversky indexes; the latter are moreover set-based and not sufficient for the vector-based setting of this paper.

Note that for two words to have a similar meaning it is not necessary that they often occur close to each other, but necessary that they have occur close to (or have) same features. For example, it is not very common that 'ship' and 'boat' occur in the vicinity of 5-words of each other, but it is indeed the case that they both often occur 5 words close to same verb, such as 'sail' and same noun modifiers, such as 'ocean'. As a result, vectors of 'ship' and 'boat' have a small geometric distance (and a high cosine value). The more same features two words w, w' have in common, the higher the corresponding coordinate on that feature will be. This will increase the inner product of word vectors $\overrightarrow{w}$ and $\overrightarrow{w}'$ and their *cosine* similarity, the latter being based on inner product.

Whereas traditional distributional models focus on meaning representations for words (and perhaps sometimes for short two-word phrases such as adj-noun), compositional versions of them are able to represent meanings of phrases and sentences by combining the vectors of the words therein. The most general such approaches are heavily based on the grammatical structure of a string of words, expressed in a typed categorial syntax such as pregroup grammars, syntactic calculus [6, 5], or Combinatorial Categorical Grammar (CCG) [24, 20]. From an abstract perspective, the vector representation of a sequence of words $\text{w}_1, \ldots, \text{w}_n$ is obtained by the following composition of functions

$$f\left((g(\text{w}_1), \ldots, g(\text{w}_n)), (x_1, \ldots, x_n)\right) \tag{1}$$

where, the meaning of a word w_i with a basic grammatical type x_i is represented by a vector in an atomic space X with a fixed basis:

$$g(\text{w}_i) = \overrightarrow{\text{w}_i} \in X$$

The meaning of a word w_j with a grammatical functional type $((x \to y) \to \cdots) \to z$ is represented by induction as a linear map $((X \to Y) \to \cdots) \to Z$, or equivalently (using the map-state duality) a matrix in the tensor space $((X \otimes Y) \otimes \cdots) \otimes Z$. We use a simplified notation and denote these by vector signs as well:

$$g(\text{w}_i) = \overrightarrow{\text{w}_i} \in ((X \otimes Y) \otimes \cdots) \otimes Z$$

Most generally, the composition function f could be the multilinear algebraic operation of tensor-contraction. In the context of this paper, we work with svo-phrases "subj verb obj". Here, basic words "subj" and "obj" are nouns with type n; their meanings are vectors in a fixed-basis atomic feature space N. Function words are transitive verbs with type $(n \to n) \to s$; this means that they input two noun phrases and output a phrase of type sentence s. The verb meanings are linear maps $(N \to N) \to S$, equivalently tensors in the space $(N \otimes N) \otimes S$. Here, f becomes matrix multiplication. The meaning of the svo-phrase "subj-verb-obj" is a vector in the sentence space S, computed by the following general formula:

$$f\left((\overrightarrow{\text{w}}_1, \overrightarrow{\text{w}}_2, \overrightarrow{\text{w}}_3), (\text{subj}, \text{verb}, \text{obj})\right) = (\overrightarrow{\text{w}}_2 \times \overrightarrow{\text{w}}_3) \times \overrightarrow{\text{w}}_1 \tag{2}$$

We shorthand the right hand side of the above as follows, identifying the notation for words with that of their grammatical types:

$$(\overrightarrow{\text{verb}} \times \overrightarrow{\text{obj}}) \times \overrightarrow{\text{subj}} \tag{3}$$

Once again, one can use the cosine similarity, this time between the phrase vectors, to measure their degree of similarity.

2.2 IR Models and Phrase-based Semantic IR

Whereas traditional IR relies on the "so-called" bag-of-words retrieval models, semantic IR deals with entities and relationships. The following queries illustrate the difference between word-only IR and semantic IR.

```
1  # Retrieve Documents in which the words
2  # peter, friend, and mary occur.
3  ?- D[ peter & friend & mary ];

5  # Retrieve Documents in which the proposition
6  # ''peter is a friend of mary" occurs.
7  ?- D[ peter.friendOf(mary) ];
```

The first query will retrieve documents in which the respective words occur. Whether there is a conjunctive or disjunctive interpretation is for the moment of secondary priority. Regarding the score contribution of the different words, all retrieval models will give more impact to rare words, and a document scores high if it has many occurrences of the query words.
The second query will retrieve all documents in which the respective *proposition* is true. The syntax applied in the examples has been introduced in [29, 11]. The probabilistic object-oriented logic (POOL) is a high-level abstraction to describe semantic retrieval and to represent knowledge. POOL is related to description logic [25] and frame-based logics [19].
The main challenge is to define a retrieval status value (RSV) that takes into account the semantic meaning of the proposition. The most naive approach would be to consider the proposition as an atomic symbol, and simply introduce something like a Proposition-Frequency and the IDF of a proposition. With this approach, a proposition is treated as a phrase (compound, sequence of words). Obviously, such an approach will be too specific in the sense that we wish to retrieve documents that contain propositions that are similar to query proposition.
Therefore, in semantic retrieval we apply a compositional approach which in essence retrieves documents for the entities and relationships of a proposition. [2, 1] introduce techniques to process and classify semantic queries. The main approach is to score documents with respect to an aggregated score that is composed of a score representing the match wrt to the subject, relationship name (or attribute name), and object (or attribute value).
To illustrate the composition of the aggregated score, consider first the simple, only word-based (term-based) TF-IDF model, where $\text{TF}(t,d)$ is the *term frequency*, and $\text{IDF}(t)$ is the inverse document frequency. The retrieval status values (RSV) is commonly defined as follows:

$$\begin{aligned}\text{RSV}_{\text{TF-IDF}}(d,q) &:= \sum_{t\in q} \text{score}(t,d,q)\\ \text{score}(t,d,q) &:= \text{TF}(t,d)\cdot\text{IDF}(t)\end{aligned}$$

The generalisation is a formulation for phrases rather than terms (words):

$$\begin{aligned}\text{RSV}_{\text{semantic-TF-IDF}}(d,q) &:= \sum_{\text{phrase}\in q} \text{score}(\text{phrase},d,q)\\ \text{score}(\text{phrase},d,q) &:= \text{\# see section 3.2}\end{aligned}$$

We refer to this approach as a *macro* score, since the aggregation is over scores that originate from a score for the respective phrase. The score of a phrase is composed of scores for the phrase components (subjects, relationships, objects, attribute names, and attribute values). Section 3.2 will expand on the approach chosen for this paper.
The following examples illustrate the difference between relationship and attributes. The phrase peter.friendOf(mary) and peter.worksFor(ibm) are subject.relship(object) propositions, whereas ibm.business_area("IT") and mary.job_title("Lecturer") are object.attrName(attrValue) propositions. For the context of the work reported in this paper, there is an additional type of proposition, namely

subject.verb(object)

where for the dataset considered, subject and object are to be understood as object types (e.g. agent, mother, land, house) rather than concrete objects. The indexing process builds indexes for subjects, relationship names, objects, attribute names, and attribute values. The retrieval process interprets a query such as "D[peter.friendOf(mary)]" and ranks the documents with a score that is based on the composition of sub-scores as described above. Though we utilise in this paper only the semantic variant of TF-IDF score, it is evident that this approach applies for any retrieval model [2].
This approach to semantic IR is powerful but poses major challenges. Firstly, the parser (for IR, usually a simple tokeniser) needs to recognise propositions, and build a space of propositions. Secondly, the query processing requires to match similar propositions (similar components), i.e. it is not sufficient to simply apply a syntactic match.
This is precisely where results of DS research contribute to solve one of the major short-comings of semantic retrieval, namely the mismatch problem between semantic propositions.

3. SIMILARITY IN NLP AND IR

Considering a subject-verb-object phrases as a source (query), in IR the task is to rank other phrases (targets) such that the ranking reflects that the target phrase is considered "relevant" with respect to the source phrase.
From a retrieval point of view, we retrieve documents that contain target phrases that are implied by the source phrase. The following example illustrates the scenario:

```
1  # 1. The case of word-based IR:
2  ?- D[ vehicle ];
3  # d123[ boat ]
4  # ontology: boat is a sub-concept of vehicle, in other
       words, boat->vehicle.
5  # Since boat implies vehicle, retrieve doc123.
7  # 2. The case of knowledge-based (semantic) IR:
8  ?- D[ person.spends(money) ];
9  # doc456[ lecturer.buys(boat) ]
10 # Since the sentence lecturer.buys(boat) implies person.
       spends(money), retrieve doc123.
11 # Note that lecturer implies person (semantic relationship
       between subjects), but
12 # boat does not semantically imply money (no semantic
       relationship between objects).
```

The DS approach would rank documents that contain phrases similar to the query higher than documents that do not have similar phrases. Distributional similarity is an appealing concept because although being symmetric [33], it is shown to correlate with human judgements [4]. In principle, one expects that it should be able to capture both ontological and semantic relationships in feature spaces: "woman" and "mother" will appear in similar context (thus share many features), and all the things that are capable of buying will co-occur with "buy" in a corpus (thus share a feature).
Whereas similarity measures are symmetric, document retrieval scores for a query are not symmetric. For example, if the query is about vehicles, then documents about both cars and bicycles are relevant. However, for a query about bicycles, not all documents about all different kinds of vehicles (e.g. cars) are relevant. Thus, similarity can also be defined through entailment, two entities (words or phrases) are similar if entailment holds in both directions.
The DS notion of similarity is defined as follows:

DEFINITION 1 (SENTENCE SIMILARITY). *The degree of similarity between two sentences s and t is a function r of their semantic representations $[\![s]\!]$ and $[\![t]\!]$, respectively obtained from the composition f of the semantic representations $[\![v_1]\!],\ldots,[\![v_m]\!]$ and $[\![w_1]\!],\ldots,[\![w_n]\!]$ of the words therein and their corresponding*

grammatical roles $x_1, \ldots, x_m$ and $y_1, \ldots, y_n$. Formally, we have:

$$\begin{aligned}\mathrm{sim}(s,t) &:= r(\llbracket s \rrbracket, \llbracket t \rrbracket)\\ &:= r(f((\llbracket v_1 \rrbracket, \ldots, \llbracket v_m \rrbracket), (x_1, \ldots, x_m)),\\ &\quad f((\llbracket w_1 \rrbracket, \ldots, \llbracket w_n \rrbracket), (y_1, \ldots, y_n)))\end{aligned}$$

In both IR and DS, the semantic representations of words and sentences are their vector representations. In DS r is the cosine of the angle and f is the composition function as defined in Equation 1. In IR, r is summation and f is the TF-IDF quantification. For both cases, in this paper we have $m = n = 3$.

The vector-based instantiation of our notion of similarity falls under the category "statistical similarity", relates to the LSA model reviewed in [15], and is an area of research studied by SemEval (a series of workshops for evaluation of computational semantic analysis systems). Traditional LSA is word-based and is more recently extended to sentences by set-based methods such as addition and multiplication. Our notion is grammar-based and leads to composition operators such as tensor contraction. SemEval datasets consider a wide variety of sentences, wheres our dataset was designed for svo phrases in order to keep the task clean and avoid the noise caused by, e.g. articles and individual entities. Further challenge would be to generalise our notion to any type of phrase.

Sentence similarity in DS is a difficult task to design and to decide. Occurrence frequencies within a corpus and semantic word hierarchies from lexical databases such as `WordNet` are used to design such tasks and human annotators are used to decide them.

When building similarity datasets, instructions are given to humans to reflect the symmetry of similarity. A participant is asked to provide a Boolean answer (yes if sentences are similar and no if sentences are not similar) or a number from a scale e.g. from 1 to 7 (1 for dissimilar and 7 for similar). To come up with similarity scores shown in Figure 1 one can compute the percentage of positive answers for Boolean answers or take the average of responses if they are on a scale or count. The number of judgements per sentence pair plays an important role in deciding the final score. For Boolean answers, that are easy to answer, one would want to collect more responses than for the scale based answers. The number of sentence pairs also plays an important role. Is it better to have judgements for more pairs, but a few Boolean answers; or for less pairs, but a lot of scale based responses?

IR evaluation datasets tend to prefer more pairs to be evaluated, despite the evaluation thoroughness, while DS datasets [27, 16, 17] tend to prefer more elaborated responses for fewer items.

Sentence 1			Sentence 2			Average
Subject	Verb	Object	Subject	Verb	Object	Similarity
			wife	pour	tea	2.696
woman	drink	water	doctor	use	test	1.125
			system	use	method	1.083
			delegate	buy	land	3.360
agent	sell	property	family	buy	home	3.125
			group	hold	meeting	1.167

Figure 1: Examples of average human annotations for svo-phrase similarity scores. Averages are over 20 judgements per sentence pair.

3.1 Vector-based Distributional Similarity

Distributional semantic models are varied over three sets of parameters: the normalisation method, the underlying corpus of text, and the composition operators. The latter differ over size and the sparsity of the information they provide about meanings of words: the larger they are, the less sparse information they contain.

The normalisation methods provide different weighting schemes for the co-occurrence counts (previously denoted by $N(f,t)$) of the word vectors. To be more explicit, the raw counts in a window of size 5 are denoted by $N5(f,t)$, the most basic normalisation method is a conditional probability, the *likelihood ratio* version of which has been shown to perform well in semantic tasks [4]. The logarithmic non-negative version of likelihood ratio, referred to by *positive point-wise mutual information* (PPMI), has been shown to perform better than *likelihood ratio* in semantic tasks [18]. These methods are summarised below:[1]

1. Raw Counts: $\mathrm{N5}(f,t)$,
2. Conditional Probability: $\mathrm{P5}(f|t) = \frac{P(f,t)}{P(t)}$,
3. Likelihood Ratio: $\mathrm{LR5}(f,t) = \frac{P(f|t)}{P(f)} = \frac{P(f,t)}{P(f)P(t)}$,
4. Positive PMI: $\mathrm{PPMI5}(f,t) = \max(0, \log(\frac{P(f|t)}{P(f)}))$.

EXAMPLE 1 (WORD LEVEL MEANING REPRESENTATION). *Given a feature (vector) space that consists of 1,000 words. Let "ocean", "sailing", "animal", "pet" be the first four words.*
Note that the feature words are independent of the source and target words. This is the main difference to the IR approach where the association between source and target is assumed to be quantified. The target vectors for "ship", "boat", "cat", and "dolphin" could be as follows:

$$\overrightarrow{ship} = (4,3,0,0,\ldots) \qquad \overrightarrow{boat} = (2,3,0,0,\ldots)$$
$$\overrightarrow{cat} = (0,0,4,3,\ldots) \qquad \overrightarrow{dolphin} = (2,1,1,0,\ldots)$$

These express that the target word "ship" occurs four times in the neighborhood of "ocean" (the feature before or after the target word) and three times in the neighborhood of 'sailing', but 0 times in the neighborhoods of "animal" and "pet" . Whereas, "cat" occurs 0 times in the neighborhoods of "ocean" and "sailing and 4 and 3 times in the neighborhoods of "animal" and "pet".
For the purpose of demonstration suppose that all the other coordinates of these ship and boat are zero, hence the lengths of their vectors become $\sqrt{25}$ and $\sqrt{13}$, and we obtain $\cos(\overrightarrow{ship}, \overrightarrow{boat}) = 0.943$; this is despite the fact that ship (as target) and boat (as source) have occurred together or not.
Let the word ship occur 20 times, and ocean and sailing occur 100 and 10 times, respectively, in the corpus the co-occurrence matrix is based upon. Then, for $P(ocean) = 100/1000$ and $P(sailing) = 10/1000$, we obtain:

$$\mathrm{N5}(ocean, ship) = 4 \qquad \mathrm{N5}(sailing, ship) = 3$$
$$\mathrm{P5}(ocean|ship) = 4/20 \qquad \mathrm{P5}(sailing|ship) = 3/20$$
$$\mathrm{LR5}(ocean, ship) = 2 \qquad \mathrm{LR5}(sailing, ship) = 15$$
$$\mathrm{PPMI5}(ocean, ship) = 0.301 \qquad \mathrm{PPMI5}(sailing, ship) = 1.176$$

A point-wise application of the above schemes to each coordinate of a target vector provides vectors for each scheme. The vectors of "ship" in each scheme is as follows:

$$\mathrm{N5} = (4,3,0,0,\ldots)$$
$$\mathrm{P5} = (4/20, 3/20, 0, 0, \ldots)$$
$$\mathrm{LR5} = (2, 15, 0, 0, \ldots)$$
$$\mathrm{PPMI5} = (0.301, 1.176, 0, 0, \ldots)$$

[1] We show the parameters for the window size 5. Any window can be chosen: 2, 5 and 10 are most common in DS.

For an svo-phrase, the most general composition operator is matrix multiplication. In this case, the distributional hypothesis over a feature space only provides us with vectors for atomic words. Hence, a crucial challenge is how to concretely build a cube (tensor of rank 3) for the verb. This area constitutes an active recent trend in DS. The most costly models thereof, generalise the original approach of [3] and use multi-step linear regression to combine feature vectors of "verb-obj" and "subj-verb" phrases [12]. A slightly more simplified approach argues for the use of matrices rather than cubes and combines the verb matrices built by linear regression from "verb-obj" and "subj-verb" phrases [28]. Less costly approaches, however, argue against the use of vector support machines at all and work with combinations of feature vectors of subjects and objects of the verb [13]. A much cheaper model which has outperformed quite a few of the above is taking the Kronecker product of the feature vector of the verb with itself [14]. Finally, the most simple setting (which has also performed a par with other models) is to work with the feature vector of the verb, where composition reduces to point-wise multiplication or addition of vectors [27]. For the purpose of this paper, we work with these and the Kronecker as our models.

1. Addition: $\overrightarrow{s} = \overrightarrow{\text{subject}} + \overrightarrow{\text{verb}} + \overrightarrow{\text{object}}$
2. Point-wise multiplication: $\overrightarrow{s} = \overrightarrow{\text{subject}} \odot \overrightarrow{\text{verb}} \odot \overrightarrow{\text{object}}$
3. Kronecker: $\overrightarrow{s} = \overrightarrow{\text{verb}} \odot (\overrightarrow{\text{subject}} \otimes \overrightarrow{\text{object}})$

The additive model is disjunctive over the information of the terms of the phrase. Its resulting vector contains all the features of subject, verb, and object, hence a bit noisy. The multiplicative model is conjunctive and consists of features that are shared between the phrases; hence too refined. The Kronecker method doubles the information of the verb into a tensor, one copy interacts with the subject, one with the object, and the results are merged, allowing the subject and object interact with each other through the verb features. This becomes evident by observing:

$$\overrightarrow{\text{verb}} \odot (\overrightarrow{\text{subject}} \otimes \overrightarrow{\text{object}}) = (\overrightarrow{\text{verb}} \odot \overrightarrow{\text{subject}}) \otimes (\overrightarrow{\text{verb}} \odot \overrightarrow{\text{object}})$$

The first $\odot$ intersects the features of verb and subject, the second one does so with the features of verb and object, the $\otimes$ puts this two together in a matrix. This lessens the noise and is not as refined.

EXAMPLE 2 (COMPOSITION). *Consider the N5 model, the sentence "dolphins follow boats", and the verb vector therein* $\overrightarrow{follow} = (1, 1, 2, 1, \ldots)$. *The additive and multiplicative vectors and the Kronecker tensor are as follows:*

$$\overrightarrow{dolphins} + \overrightarrow{follow} + \overrightarrow{boats} = (5, 5, 3, 1, \ldots)$$

$$\overrightarrow{dolphins} \odot \overrightarrow{follow} \odot \overrightarrow{boats} = (4, 3, 0, 0, \ldots)$$

$$\overrightarrow{follow} \odot (\overrightarrow{dolphins} \otimes \overrightarrow{boats}) = \begin{pmatrix} 4 & 6 & 0 & 0 & \ldots \\ 2 & 3 & 0 & 0 & \ldots \\ 4 & 6 & 0 & 0 & \ldots \\ 0 & 0 & 0 & 0 & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 1 & 2 & 1 & \ldots \\ 1 & 1 & 2 & 1 & \ldots \\ 2 & 2 & 4 & 2 & \ldots \\ 1 & 1 & 2 & 1 & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots \end{pmatrix} \odot \begin{pmatrix} 4 & 6 & 0 & 0 & \ldots \\ 2 & 3 & 0 & 0 & \ldots \\ 2 & 3 & 0 & 0 & \ldots \\ 0 & 0 & 0 & 0 & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots \end{pmatrix}$$

The additive vector has non-zero coordinates on "ocean", "sailing", and "animal", i.e. the features of "dolphins", "follow", and "boats". The multiplicative only has non-zero features on "ocean" and "sailing". The Kronecker has pairs of features of all the terms, thus relating their non-zero coordinates, e.g. "(animal, ocean)" relates features of "dolphin" to that of "boat", whereas "boat" had originally a zero coordinate on "animal", making the two interact with each other through the non-zero "animal" feature of "follow".

3.2 Phrase-based Semantic IR

The main strands of IR models (TF-IDF, BM25, LM) can be related to the measuring the dependence between document (source) and query (target). The document-query independence (DQI) measure is as follows [30]:

$$\text{DQI}(d, q) := \log \frac{P(d, q)}{P(d) \cdot P(q)} \tag{4}$$

This component occurs for a distribution of documents and queries in the formulation of the mutual information $\text{MI}(D, Q)$, and is also referred to as point-wise mutual information ($\text{DQI}(d, q) = \text{pmi}(d, q)$).
Depending on the decomposition of the participating probabilities, one can derive TF-IDF or LM. TF-IDF for $P(d|q)/P(d)$ and LM for $P(q|d)/P(q)$. For the purpose of this paper, we focus on the TF-IDF-side of retrieval, and we choose the disjunctive decomposition of $P(q|d)$ over a space of disjoint terms.
For $P(q)$ being constant when ranking documents, one obtains:

$$P(q|d) = \sum_t P(q|t) \cdot P(t|d)$$

Herein, $P(q|t)$ can be represented by the IDF ($P(q|t)$ high for rare terms), and $P(t|d)$ is proportional to the TF quantification. This leads to the TF-IDF score:

$$\text{RSV}_{\text{TF-IDF}}(d, q) = \sum_{t \in q} \text{IDF}(t) \cdot \text{TF}(t, d)$$

BM25 (without relevance information) is basically TF-IDF where the TF quantification becomes $\text{tf}_d/(\text{tf}_d + 1)$ for a document of average length (omitting BM25-TF parameters such as k_1 and b).
In the following, this probabilistic derivation and formulation of TF-IDF is extended with respect to:

1. Generalised TF-IDF score for matching subject-verb-object phrases.
2. Relate query concept (e.g. subject) to document concept since the concepts may be different (no symbolic match).

Let svo_d and svo_q be subject-verb-object phrases, where d is the source, and q is the target. The source phrase is denoted as $\text{svo}_d = s_d, v_d, o_d$, and the target phrase is $\text{svo}_q = s_q, v_q, o_q$.
For the IR-based models, we require a notion of TF, i.e. the degree to which a query subject, relationship/verb and object represents a virtual document. For example, for the query "sailor enjoys trip", we want to retrieve documents that contain "dolphins follow boat". For each component of the query phrase, we need an estimate of $P(t_q|d)$, where t_q is the component of the query phrase, and d is a virtual document that contains the target phrase.
Such estimate can be obtained via a sampling (voting). Then, the sampling over a space of disjoint concepts is:

$$P(t_q|d) = \sum_{t \in T} P(t_q|t) \cdot P(t|d)$$

Here, T is a space of concepts and the concepts are assumed to be disjoint events. The probability $P(t_q|t)$ is estimated based on co-occurrence information. For example:

$$P(t_q|t) = P(s_q|s_d) = P(\text{sailor}|\text{dolphins}) = \frac{\text{N5}(\text{sailor}, \text{dolphins})}{\sum_x \text{N5}(x, \text{dolphins})}$$

With regard to logical and probabilistic retrieval, we are utilising this information into a model based on the probability that the source document implies the target query.

$P(d \rightarrow q) = P(\text{dolphins, follow, boat} \rightarrow \text{sailor, enjoys, trip})$

Target w_t	Source w_s	N5	$P5(w_t\|w_s)$
sailor	dolphins	100	100/1,000
dolphins	sailor	100	100/2,000
trip	boat	500	500/3,000
boat	trip	500	500/4,000
follow	enjoy	800	800/8,000
enjoy	follow	800	800/10,000

In addition, for each word there are the usual statistics such as the number of times the word occurs. This may include the number of locations $N_{\text{Locations}}(w)$, the number of documents $N_{\text{Documents}}(w)$, the number of sentences, the number of document titles, etc. For the purpose of this work, it is sufficient to consider the main two counts, number of locations and number of documents.
For example, sailor occurs $2,000$ times over $1,000$ documents (avgtf(sailor) $= 2$) and dolphins occurs $1,000$ times over 800 documents (avgtf(dolphins) $= 1.25$). For illustrating the score computation, assume the following statistics:

word w	$N_{\text{Locations}}(w)$	$N_{\text{Documents}}(w)$
dolphins	1,000	800
sailor	2,000	1,000
boat	3,000	1,500
trip	4,000	2,000
enjoy	8,000	3,000
follow	10,000	4,000

Let there be 10^6 documents in total.

EXAMPLE 3 (SCORE COMPUTATION). *Let d and q be svo-phrases. Then, the probability of the implication is estimated via the conditional probability, which is estimated via the total probability theorem.*

$$P(d \rightarrow q) = P(q|d) = \sum_{x \in \{subject, verb, object\}} P(q|d, x) \cdot P(x)$$

Here, x is the type *of the component, i.e. subject, verb or object. The sample over a space of word types breaks up the phrase into its components, and applies type-specific probabilities. For example, for the subject:*

$$P(q|d, subject) = P(sailor|d, subject)$$

The implication dolphin $\rightarrow$ sailor can be viewed as a translation task. A term t_q is assigned to a document d if the term is similar to many terms that occur in d. This estimate is based on the total probability:

$$P(t_q|d) = \sum_{t'} P(t_q|t') \cdot P(t'|d)$$

Since the virtual document d contains exactly one phrase, $P(t'|d) = 1$. This would be more general for a real retrieval task where q and d are sets of phrases.
For the query example "sailor enjoys trip", where the document contains "dolphins follow boat", for the subject, the following expression show-cases the estimation of $P(t_q|d, subject)$.

$$P(sailor|d, subject) = P(sailor|dolphins) \cdot P(dolphins|d, subject)$$

For the subject-subject, verb-verb, and object-object probabilities, we have:

$$P(sailor|dolphins) = 100/2,000 = 0.05$$
$$P(enjoys|follow) = 800/8,000 = 0.1$$
$$P(trip|boat) = 500/4,000 = 0.125$$

This leads to the following score reflecting the inference between the two svo-phrases:

$$\begin{aligned} score(d, q) = & \\ = & \; P(s_q|d) \cdot IDF(s_q) + P(v_q|d) \cdot IDF(v_q) + P(o_q|d) \cdot IDF(o_q) \\ = & \; 0.05 \cdot \log \frac{10^6}{1,000} + 0.1 \cdot \log \frac{10^6}{3,000} + 0.125 \cdot \log \frac{10^6}{2,000} \end{aligned}$$

The example illustrates clearly that the score is not symmetric; this is evident because $P(t_q|d)$ is different from $P(t_d|q)$. Also, the IDF component is applied to the query components.
The score is high if the components of the query phrases are related (modelled by $P(t_q|d)$) and the query component is rare (modelling by IDF(t_q)).
Note that in this approach we consider subject, verb and object as *disjoint* events. A more general approach could consider the relationship between subject-verb and verb-object and subject-object.
Moreover, the IDF could be specific with respect to the component type, i.e. the IDF of a subject is different from the IDF of a verb. This issue becomes evident for words that are both, noun and verb, such as "help", "need" and "demand".
Depending on the co-occurrence information chosen, that is 1) N5, 2) P5, 3) LR5, and 4) PPMI5, the estimate of $P(t_q|d)$ is different, and this leads to four different candidate models.
In the evaluation we measure the performance of different instantiations of this semantic TF-IDF model where we vary the way the IDF is defined, and, more importantly, we vary the way the TF component is derived from the co-occurrence information.
The main questions we investigate are:

1. How does the score described for the compositional TF-IDF compare to the DS approach computing phrase similarity via distributional (co-occurrence) representations?

2. How sensitive are the approaches with respect to the data feed? The quality of the co-occurrence information between symbols is essential for IR, whereas DS relies on the indirect measure.

4. NLP MEETS IR

The DS approach uses a high-dimensional feature term space to measure the similarity between terms. The primary events of this space are the target and features terms. The meaning of a term in this space is represented by a vector of its features and the meaning of a document is a sequence of its terms. This is different from conventional IR approaches [8] that are based on a co-occurrence quantification such as the number of documents both terms occur in, or approaches in LM that measure the probability that two terms are related. In these approaches, the primary events are the document and the query and the meaning of a term is just an atomic symbolic entity. The model does not represent the meaning of a term in any form and it is only concerned about whether it occurs with another symbol or inside a document. The correspondence between the IR and DS approaches is summarized in the Table 1.
The combination of the two approaches enables us to put together the symbolic meanings of terms in a foreground IR model (both in a query and in a document) with vector representations coming

	Distributional Semantics	Information Retrieval
Co-occurrence	Co-occurrence between the semantic symbol (target word) and feature words.	Co-occurrence between the semantic symbols (words) themselves.
Representation of words	Distributional.	Symbolic.
Single vs set	Similarity between two *single phrases*.	Relationship (implication, entailment) between two *sets of phrases*.
Relationship symmetry	Similarity is a symmetric function.	Relationship between sets is not symmetric; moreover, the phrase-based score is not necessarily symmetric.
Similar/relevant	Phrase t_i is similar to phrase t_j.	Document d (source) is relevant with respect to query q (target).
Scores	The similarity score is estimated based on the distance/angle between distributional vectors.	The relevance score is estimated based on the retrieval model that computes the implication between the set of document and the set of query propositions.
Probabilistic semantics	(In)dependence between a target word and a feature word: $\frac{P(w_t,w_f)}{P(w_t)\cdot P(w_f)}$.	(In)dependence between a document and a query: $\frac{P(d,q)}{P(d)\cdot P(q)}$.
In this work: virtual query has exactly one proposition; virtual document has exactly one proposition; therefore, the similarity score $\mathrm{sim}(\textit{phrase1}, \textit{phrase2})$ *can be compared to the retrieval score* $\mathrm{RSV}(\textit{document: set of phrases}, \textit{query: set of phrases})$.		

Table 1: Distributional semantics and IR side by side. DS focuses on the similarity between two target words and phrases; IR focuses on the similarity between a source document and a query document.

from a background DS model. A main consequence of the internal representations of DS is that these models are able to utilise an intermediate representation, namely the co-occurrence quantification of term over a space of feature terms, whereas the conventional IR approach relies on the explicit relationship between terms. One could embed the intermediate representation into the IR model by forming a λ-mixture model: keep λ percent of the co-occurrence quantification from the IR model $P_{fg}(s_q \mid s_d)$ and substitute the other $1-\lambda$ percent with a quantification over the degree of similarity between the DS model $P_{bg}(s_q \mid s_d)$:

$$P_{\lambda\text{mix}}(s_q \mid s_d, \text{fg}) = \lambda P_{\text{fg}}(s_q \mid s_d) + (1-\lambda) P_{\text{bg}}(s_q \mid s_d) \quad (5)$$

We refer to this variant by DS-based IR or Symb&Distr.
To illustrate, instantiate $\lambda = 0.5$. Consider two sentences "men love yachts" and "women love porsches" with data as follows (usual probabilities for the IR model and the cosine distance for the DS model):

$$P_{\text{IR}}(\text{men}|\text{women}) = 0.2 \qquad P_{\text{IR}}(\text{yachts}|\text{porsches}) = 0.1$$
$$P_{\text{DS}}(\text{men}, \text{women}) = 0.72 \qquad P_{\text{DS}}(\text{yachts}, \text{porsches}) = 0.80$$

where "yachts" occurred a total of 1,016 times in the corpus, "porsches" 259 times, "love" 22,348 times, "men" 37,007 times (data from BNC). The weights of the DS-based IR model will be:

$$P_{\lambda\text{mix}}(\text{men}|\text{women}, \text{sim}) = \frac{0.2}{2} + \frac{0.72}{2}$$
$$P_{\lambda\text{mix}}(\text{yachts}|\text{porches}, \text{sim}) = \frac{0.1}{2} + \frac{0.80}{2}$$

In the absence of number of documents from the data corpora of DS, we work with the number of times a term occurred in the whole corpus (taking this assumption in an IR setting means we are assuming that the term occurred one time in each document). Hence, $P(t_q \mid d)$ becomes the total number of terms in the corpus (in this example 10^8) divided by the number of times t occurred in the corpus. The IR score between the sentences is now computed as follows:

$$\left(\frac{0.2}{2} + \frac{0.72}{2}\right) \cdot \log \frac{10^8}{37,007} + 1 \cdot \log \frac{10^8}{22,348} + \left(\frac{0.1}{2} + \frac{0.80}{2}\right) \cdot \log \frac{10^8}{1,016}$$

The mixture model yields a higher score than the IR-only model. This is because the similarity based background DS model has higher values (e.g. 0.72) than the IR model direct occurrence (e.g. 0.2). The IR model however performs surprisingly well on its own.
From an IR/LM perspective, there is only a relationship between yachts and porsches if the words (symbols) occur together in some context. From a DS perspective, the relationship is established over the co-occurrence with feature terms, e.g. "expensive" and "run". Since boats and cars occur in the context of the same feature terms, a semantic relationship is established.
Furthermore, DS is phrase-oriented, whereas in IR, the bag-of-words approach is still dominant. The standard IR models are formulated for words, there is no agreed standard for semantic and/or compositional models.

5. EVALUATION

We evaluate models by the correlation between model score and reference score. For DS models this is similarity based, for IR models, we have a simulation where a query contains exactly one phrase and a document also contains only one phrase.
We evaluate models described in Section 3 on a document scoring task. Concretely, given a query and several candidate documents, a model has to assign scores to candidates. The average score correlation per query is measured.

5.1 Evaluation Dataset

The transitive sentence similarity dataset[2] described in [17, 16] is used. It consists of 108 subject-verb-object subject-verb-object phrase pairs, some of them are shown on Figure 1. There are 2,603 reference similarity scores (scale 1-7, 1 for least similar, 7 for most similar). These correspond to 20-25 reference scores per pair. To have a unique score per pair, we use the average of human judgements as the gold standard. Hereby, the reference score distribution is based on the dataset where each test phrase is paired with three other which represent a relatively similar, a medium similar, and a least similar candidates.

[2] http://www.cs.ox.ac.uk/activities/compdistmeaning/emnlp2013_turk.txt

Hyper Parameters		DS Models							IR Models			
Corpus	Quantification	Addition	Verb		Multiplication		Kronecker		Symbolic		Symb&Distr	
BNC	N5	0.372	0.380	+0.023	0.107	-0.712†	0.159	-0.572*	0.559	+0.505*	0.593	+0.596†
	P5	0.555	0.380	-0.316*	0.107	-0.808‡	0.159	-0.714‡	0.573	+0.033	0.637	+0.148
	PPMI5	0.704	0.412	-0.415‡	0.701	-0.004	0.680	-0.034	0.674	-0.043	0.725	+0.029
	LR5	0.660	0.648	-0.018	0.773	+0.171	0.826	+0.251‡	0.699	+0.059	0.724	+0.097
ukWaC	N5	0.495	0.425	-0.142	0.212	-0.572‡	0.167	-0.663‡	0.649	+0.311*	0.698	+0.411†
	P5	0.632	0.425	-0.329†	0.212	-0.665‡	0.167	-0.736‡	0.639	+0.010	0.769	+0.216†
	PPMI5	0.741	0.496	-0.331‡	0.770	+0.038	0.766	+0.033	0.742	+0.001	0.779	+0.051
	LR5	0.724	0.561	-0.226*	**0.841**	**+0.161***	**0.861**	**+0.189**†	0.767	+0.058	0.785	+0.083

Figure 2: Average score (Pearson) correlation per query. Signed numbers show a relative improvement with respect to the baseline (Addition). Average relative increase from Symbolic IR models to Mixed IR models is 0.068 for BNC and 0.063 for ukWaC. *statistically significant under $p < 0.1$. †s.s. under $p < 0.05$, ‡s.s. under $p < 0.01$. T-test of two independent samples.

We assume that in the retrieval scenario similarity scores can be converted to retrieval score rankings, in other words, for each query phrase its most similar counterpart is the most relevant document, the somewhat similar counterpart is the second relevant document, and, finally, the dissimilar sentence is the least relevant document.

5.2 Model Hyper Parameters

Two corpora are used to obtain co-occurrence counts for the DS feature vector spaces and the IR term relatedness scores. The British National Corpus (BNC)[3] [7] contains both written and spoken language and consists of 200 million words. Currently, BNC is considered to be a small resource to build distributional vector space models. Instead, ukWaC[4] [9], a 2 billion word collection of `dot-uk` web-pages, is used. In our experiments we used PukWac (a dependency parsed version of ukWaC). Source corpus is one of the model hyper parameters that can influence model performance. In general it is expected that the larger is the corpus the more reliable are the model results.

During the co-occurrence matrix construction lemmatized versions of words were gathered[5]. Both target words and feature words were part of speech tagged, so there were different vectors for the verb "help" and the noun "help". 2000 most frequent nouns, verbs, adjectives and adverbs in a corresponding corpus formed the features. In all our experiments we used a symmetric window of 5 words.

The second hyper parameter is the quantification of the co-occurrence counts. We start off with the simplest quantification of target-feature relatedness which is the raw co-occurrence count (referred as N5). The P5 quantification is the conditional probability of feature given target.

We have two models that express feature-target (in)dependence, namely PPMI5, which is the PPMI score, and LR5, which is the likelihood ratio (refer to Section 3 for more details). Research performed on the original sentence similarity task [17, 16] reports results of more elaborated vector spaces (such as neural word embeddings and spaces based on dimensionality reduction) that are comparable to the numbers we observed for PPMI5 and LR5 [26]; that is why in this paper we only use these models.

We evaluate 8 hyper parameter combinations based on 2 corpora (BNC and ukWaC) and 4 quantification methods (N5, P5, PPMI5 and LR5).

[3] `http://www.natcorp.ox.ac.uk/`

[4] `http://wacky.sslmit.unibo.it/`

[5] For PukWaC lemmas were additionally lower-cased, as we noticed quite a lot of nouns that start with a capital latter, for example "University" when it is a part of university's name.

5.3 Candidate Models

Based on the hyper parameters and the aim to compare DS versus IR models we test 4 DS models and 2 IR models. On the DS side we experiment with models based on various compositional methods, on the IR side we contrast symbolic estimate $P(t_q|d)$ (see Section 3.2) with a mixture of symbolic and distributional evidence $P_{0.5\text{mix}}(t_q \mid d, \text{sim})$ (see Section 4, Equation 5).

We treat DS Addition as the baseline in the experiments as it is the most straightforward model. To control whether composition is done competitively, we introduce model Verb that does not do any composition and considers only the verb component of an svo-phrase. The argument here is that a verb contains a significant part of the phrase meaning and a good compositional method should not distract this signal when composing with the subject and object. Overall, we test 4 DS models (Addition, Verb, Multiplication, Kronecker) and 2 IR models (Symbolic and DS-based).

6. RESULTS AND ANALYSIS

The results are presented in Table 2 and are depicted in Figure 3. Regarding the DS models, the Kronecker model with an LR5 vector performed best in both BNC and ukWaC. In IR, the Dist&Symb model with LR5 did slightly better than the Symbolic IR with LR5. The non-compositional verb-only model was not very low, although it was below the additive baseline. The LR5 model of all of the other compositional DS and IR models increased the LR5 of the additive baseline.

In general, LR5 worked better than the N5 and P5 in both DS and IR models. This is because the LR5 is a non-logarithmic version of the *independence* measure discussed in Equation 4, hence it is a measure aware of the independence between the target and source terms and the vector components provided thereof have a better quality. Both N5 and P5 lack the explicit independence-aware expression. In the same lines, the P5 weighting measure of IR does not work so well on its own, as it is lacking independence

All models got a boost by moving from N5 and P5 to LR5 and PPMI5. This boost was particularly high in the DS multiplication model, since the introduction of likelihood ratios and logarithms decrease the eliminating effect of close-to-zero raw coordinates and probabilities. For the same reason, the amount of increase was not as much in the additive and verb-only models. This might be overcome by multiplying it with an explicit IDF component. A further observation is that despite the previous word-level DS predictions [18] PPMI5 did not do as well as LR5. This could be caused by the effect of using cosines together with composition operators, a combination seemingly less sensitive to logarithmic manipulations.

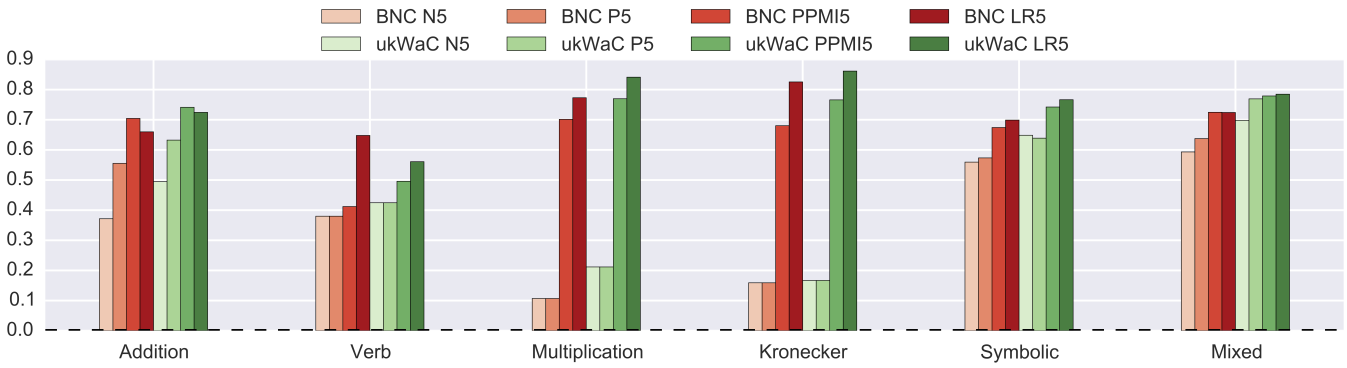

Figure 3: Graphical representation of Average score (Pearson) correlation per query. All models got a boost by using a larger corpus and the LR5 weighting. The symbolic IR model got an extra improvement when mixed with background NLP data. For each model hyper parameters are (listed from left to right): BNC N5, BNC P5, BNC PPMI5, BNC LR5, ukWaC N5, ukWaC P5, ukWaC PPMI5, ukWaC LR5.

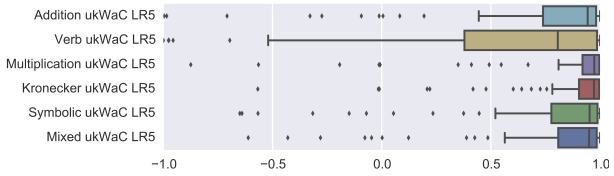

Figure 4: Distribution boxplot of query score correlations for each model. It shows distributions of query score correlations for each model, based on LR5 quantification and ukWaC corpus. All models (except Verb) got a distribution with lower standard deviation than Addition.

The best DS model was closely followed by multiplication. Figure 4 shows a more detailed analysis of this situation. We observe that the inter-quartile range of Multiplication is less than that of Kronecker. This means that Multiplication is a less conservative composition operator: it gives higher results than Kronecker on its good queries, but makes more severe mistakes on the more complex ones. The average lower performance of Multiplication is depicted by the larger number of its extreme outliers.

Somewhat unexpectedly, the original averages of the IR model (in comparison to the Symb&Distr IR) are already quite high. The involvement of explicit cosine similarities from the background DS model to the foreground IR model had a relatively small increase of 0.068 for BNC and 0.063 for ukWaC. The stability of the amount of this increase across the two corpora (despite their huge size difference) shows that a bigger corpus does not fully solve the sparsity problem of Symbolic IR. The added cosine terms, nonetheless, caused dramatic increases in particular individuals where the correlations were low (even negative). The best example herein is "case require attention", which was given a negative correlation of -0.14 by the Symbolic IR model, whereas the Symb&Distr IR model increased that by 70% to 0.56. Even more so, Symbolic IR ranked "member attend conference" as the highest match, whereas the correct option was "patient need treatment", correctly ranked so by the integrated model. Understanding which kinds of individuals lend themselves to such improvements is a direction for future work.

7. SUMMARY AND CONCLUSIONS

In this paper, we have presented the comparison of two principle methodologies to measure the similarity between semantic phrases. In particular, subj-verb-object phrases were presented, though the methodologies are not restricted to this special case.

The methodologies differ in word meaning representation and come from the fields of NLP and IR: in the DS domain of NLP, words are represented as vectors in a space of feature terms where each vector component reflects a co-occurrence quantification; in phrase-based IR, there is no internal representation of terms other than their mutual degree of co-occurrence.

The match between phrases is based on compositional models. For DS, a composition of subject, verb and object vectors represents the phrase, and the similarity between phrases is measured by cosine. For phrase-based IR, we showed a methodology to compute a score that reflects the degree to which a source phrase implies a target phrase. A source phrase corresponds to a proposition that occurs in the query, and a target phrase corresponds to a proposition that occurs in the document.

This paper tackles the term mismatch problem in information retrieval with regards to documents without direct occurrences of query terms in them. We present how this problem can be solved by using symbolic co-occurrence information. In addition, we experiment with how mixing the symbolic model with the similarity model improves the result. One of the major findings is that the DS+IR model's improvement over the IR model is less than expected, which means that the IR model score computation utilises the symbolic model to its maximum. On the other hand, the result of the NLP grammatical approach shows that there is a potential improvement if the IR models were more grammar aware.

In IR, the relevance of the source with respect to the target is based on the availability of information about the direct co-occurrence of source components (e.g. subject) and target components. The score reflecting the relevance should therefore be sensitive to the amount and quality of semantic information available. Our investigation shows that this improvement was small in average but substantial for individual cases. The impact for NLP is the finding that the more grammatically oriented composition operators, in our case the Kronecker model, provides better scores. This offers another impact to IR, which has a more set-based view on the components of phrases and should be studied in more details in the future. On the NLP side, the future work would be to look into similarity between sets of phrases, namely documents, and explore component-wise relevance measures between phrase terms. Looking at real data for the original phrase comparisons of the introduction supports this point. Namely, "agent sells property" is ranked the closest to "delegate buys land" and second closest to "family buys home". This is because "agent" is more similar to "delegate" than to "family", and "property" is more similar to "land" than to "home".

The work of this paper suggests that the NLP-based signal about the word co-occurrence is essential to perform semantic IR tasks. It also shows that the IR retrieval score, which is not a similarity measure by itself, correlates reasonably well with similarity reference scores. To show the effectiveness of the proposed approach, the method should be evaluated on a standard IR benchmark. Analytical investigation into dualities between the vector-based NLP approach to represent semantics and probabilistic IR models, including TF-IDF, Language Modelling and BM25 is another direction of extending this work. Concluding, NLP methodologies are highly beneficial for semantic IR tasks.

8. ACKNOWLEDGEMENTS

We thank the anonymous reviewers for their comments. Support from EPSRC grant EP/F042728/1 is gratefully acknowledged.

9. REFERENCES

[1] Hany Azzam and Thomas Roelleke. SQR: a semantic query rating scheme. In *Proceedings of the third workshop on Exploiting semantic annotations in information retrieval*, ESAIR '10, pages 21–22, New York, NY, USA, 2010. ACM.

[2] Hany Azzam, Sirvan Yahyaei, Marco Bonzanini, and Thomas Roelleke. A schema-driven approach for knowledge-oriented retrieval and query formulation. In *Proceedings of the Third International Workshop on Keyword Search on Structured Data*, KEYS '12, pages 39–46, New York, NY, USA, 2012. ACM.

[3] Marco Baroni and Roberto Zamparelli. Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 1183–1193, Stroudsburg, PA, USA, 2010. ACL.

[4] John A. Bullinaria and Joseph P. Levy. Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods*, pages 510–526, 2007.

[5] Bob Coecke, Edward Grefenstette, and Mehrnoosh Sadrzadeh. Lambek vs. lambek: Functorial vector space semantics and string diagrams for lambek calculus. *Ann. Pure Appl. Logic*, 164(11):1079–1100, 2013.

[6] Bob Coecke, Mehrnoosh Sadrzadeh, and Stephen Clark. Mathematical foundations for a compositional distributional model of meaning. *CoRR*, abs/1003.4394, 2010.

[7] Steve Crowdy. The BNC spoken corpus. *Leech et al*, pages 224–235, 1995.

[8] Susan T. Dumais, George W. Furnas, Thomas K. Landauer, Scott Deerwester, and Richard Harshman. Using latent semantic analysis to improve access to textual information. In *In Proceedings of Computer Human Interaction '88*, pages 281–285. ACM Press, 1988.

[9] Adriano Ferraresi, Eros Zanchetta, Marco Baroni, and Silvia Bernardini. Introducing and evaluating ukWaC, a very large web-derived corpus of English. In *Proceedings of the 4th Web as Corpus Workshop (WAC-4) Can we beat Google*, pages 47–54, 2008.

[10] John R. Firth. A Synopsis of Linguistic Theory, 1930-1955. *Studies in Linguistic Analysis*, pages 1–32, 1957.

[11] N. Fuhr, N. Gövert, and Th. Roelleke. Dolores: A system for logic-based retrieval of multimedia objects. In W. Bruce Croft, Alistair Moffat, C. J. van Rijsbergen, Ross Wilkinson, and Justin Zobel, editors, *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 257–265, New York, 1998. ACM.

[12] Edward Grefenstette, Georgiana Dinu, Yao-Zhong Zhang, Mehrnoosh Sadrzadeh, and Marco Baroni. Multi-step regression learning for compositional distributional semantics. *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013)*, 2013.

[13] Edward Grefenstette and Mehrnoosh Sadrzadeh. Experimental support for a categorical compositional distributional model of meaning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1394–1404. ACL, 2011.

[14] Edward Grefenstette and Mehrnoosh Sadrzadeh. Experimenting with transitive verbs in a DisCoCat. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, pages 62–66, Edinburgh, UK, July 2011. ACL.

[15] Montmain J. Harispe S., Ranwez S. Janaqi S. Semantic similarity from natural language and ontology analysis. *Synthesis Lectures on Human Language Technologies*, 8(1):1–254, 2015.

[16] Dimitri Kartsaklis and Mehrnoosh Sadrzadeh. Prior disambiguation of word tensors for constructing sentence vectors. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNL)*, Seattle, USA, October 2013. ACL.

[17] Dimitri Kartsaklis and Mehrnoosh Sadrzadeh. A study of entanglement in a categorical framework of natural language. In *Proceedings of the 11th Workshop on Quantum Physics and Logic (QPL)*, Kyoto, Japan, June 2014.

[18] Douwe Kiela and Stephen Clark. A systematic study of semantic vector space model parameters. In *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)*, pages 21–30, Gothenburg, Sweden, April 2014. ACL.

[19] F. Kifer and G. Lausen. F-logic: A higher-order language for reasoning about objects, inheritance, and scheme. In *Proceedings of the ACM SIGMOD International Conference on the Management of Data*, pages 134–146, New York, 1989.

[20] Jayant Krishnamurthy and Tom Mitchell. *Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality*, chapter Vector Space Semantic Parsing: A Framework for Compositional Vector Space Models. ACL, 2013.

[21] T. Landauer and S. Dumais. A Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquision, Induction, and Representation of Knowledge. *Psychological Review*, 1997.

[22] D. Lin. Automatic retrieval and clustering of similar words. In *Proceedings of the 17th international conference on Computational linguistics-Volume 2*, pages 768–774. ACL, 1998.

[23] K. Lund and C. Burgess. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods Instruments and Computers*, 28(2):203–208, 1996.

[24] Jean Maillard, Stephen Clark, and Edward Grefenstette. A type-driven tensor-based semantics for ccg. *EACL 2014 Type Theory and Natural Language Semantics Workshop*, 2014.

[25] C. Meghini, F. Sebastiani, U. Straccia, and C. Thanos. A model of information retrieval based on a terminological logic. In R. Korfhage, E. Rasmussen, and P. Willett, editors, *Proceedings of the Sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 298–308, New York, 1993. ACM.

[26] Dmitrijs Milajevs, Dimitri Kartsaklis, Mehrnoosh Sadrzadeh, and Matthew Purver. Evaluating neural word representations in tensor-based compositional settings. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 708–719, Doha, Qatar, October 2014. ACL.

[27] Jeff Mitchell and Mirella Lapata. Composition in distributional models of semantics. *Cognitive Science*, 34(8):1388–1439, 2010.

[28] Denis Paperno, The Nghia Pham, and Marco Baroni. A practical and linguistically-motivated approach to compositional distributional semantics. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 90–99. Association for Computational Linguistics, 2014.

[29] T. Roelleke and N. Fuhr. Retrieval of complex objects using a four-valued logic. In H.-P. Frei, D. Harmann, P. Schäuble, and R. Wilkinson, editors, *Proceedings of the 19th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 206–214, New York, 1996. ACM.

[30] Thomas Roelleke and Jun Wang. Tf-idf undercovered: A study of theories and probabilities. In *SIGIR*, 2008.

[31] H. Rubenstein and J.B. Goodenough. Contextual Correlates of Synonymy. *Communications of the ACM*, 8(10):627–633, 1965.

[32] H. Schütze. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–123, 1998.

[33] Peter D Turney, Patrick Pantel, et al. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37(1):141–188, 2010.