Computing and Informatics, Vol. 38, 2019, 240–264, doi: 10.31577/cai_2019_1_240

FROM PARSED CORPORA TO SEMANTICALLY RELATED VERBS

Hasan Zafari

Department of Computer Engineering
Malayer Branch, Islamic Azad University
Malayer, Iran
e-mail: Hasan.Zafari@gmail.com

Maryam Hourali

Department of Information and Communication Technology (ICT)
Malek-Ashtar University of Technology
Tehran, Iran
e-mail: mhourali@mut.ac.ir

Abstract. A comprehensive repository of semantic relations between verbs is of great importance in supporting a large area of natural language applications. The aim of this paper is to automatically generate a repository of semantic relations between verb pairs using Distributional Memory (DM), a state-of-the-art framework for distributional semantics. The main idea of our method is to exploit relationships that are expressed through prepositions between a verbal and a nominal event in text to extract semantically related events. Then using these prepositions, we derive relation types including causal, temporal, comparison, and expansion. The result of our study leads to the construction of a resource for semantic relations, which consists of pairs of verbs associated with their probable arguments and significance scores based on our measures. Experimental evaluations show promising results on the task of extracting and categorising semantic relations between verbs.

Keywords: Semantically related verbs, temporal relations, cause-effect relations, related events knowledge base

1 INTRODUCTION

Understanding the semantics relations between events is an important step to capture the meanings of text. This information is crucial in supporting textual inferences, where systems need to automatically infer unknown fact from the currently available facts. For example, consider the following snippet. In Coreference Resolution, it may be useful to know *shot* could result in *killing*.

> An Indiana teenager killed a 73-year-old man.
> Autopsy results show Kim was shot three times.

Knowing that *kill* and *shot* are semantically related, we can easily co-refer *Kim* to *a 73-year-old man* rather *An Indiana teenager* which is not its correct antecedent.

A repository that includes relationship between plausible semantically related events could be helpful in many NLP tasks, including Question Answering [1, 2], Machine Translation, Information Extraction, Coreference Resolution [3], Prediction [4], Summarization [5], Recognizing Textual Entailment [6], etc.

Many semantic resources have been developed to express knowledge of verbs, including FrameNet [7], VerbNet [8], ProBank [9], and WordNet [10]. Despite usefulness in many aspects, unfortunately these resources do not provide (large-scale) semantic knowledge between verb pairs. WordNet [10] provides some types of this knowledge as cause and entailment relations. However this information is just provided for relations that are always true. For instance, it does not include the relation "shot" happens-before "kill" since it is just a plausible sequence of events and is not guaranteed to occur all the time. Such relations hold between a wide number of event pairs but are not accessible easily.

Since a real application demands a wide-coverage resource for related verbs, we develop an automatic method to acquire a broad-coverage repository of semantic relations between verbs based on Distributional Memory (DM) [11]. The main contributions of our research are as follows:

- Providing a broad coverage Knowledge Base (KB) of semantically related verbs.
- Proposing a set of novel metrics to measure the strength of the semantically related verbs.
- Using preposition between verb-noun pairs to infer semantic relations and direction between them.
- Providing plausible common arguments of related verbs that beside other benefits, is helpful in identifying correct sense of verbs that causes their relations.
- Role mapping for the common argument of each related verb pair.

This paper is organized as follows. Section 2 describes previous attempts to discover related verbs. In Section 3 we present Distributional Memory, which is the base of our method. The model for the extraction of semantically related verbs and classification relations types is presented in Section 4, followed by the evaluation and discussion in Section 5 and conclusion in Section 6.

2 RELATED WORK

Due to importance of verb knowledge in natural language processing, many semantic resources have been developed to express knowledge of verbs, including FrameNet [7], VerbNet [8], ProBank [9], and WordNet [10]. Despite usefulness in many aspects, unfortunately these resources do not provide *broad-coverage* semantic knowledge between verb pairs, but provide information about the semantic classes, thematic roles and selectional restrictions of verbs. Among these, WordNet and FrameNet are the only resources which provide information about semantic relation between verbs. However, as these resources are created manually they have a very limited coverage. Researchers have recently shown more interest in the task of automatic recognition of causal-temporal relations between events [12, 13, 14, 15, 16, 17, 18, 19, 20, 21].

In [12] the authors use Naive Bayes classifier to learn the probabilities of semantic relation between event pair from a raw corpus in an unsupervised manner. To evaluate their model, they used two test sets from different domains. Test sets were manually classified with two human annotators. They stated that their best model improved by 7.05 % from the baseline model.

In [22, 23] the authors tried to extract chains of events sharing a common participant. They consider only verbs as events and given an existing chain of events, they predict the next likely event involving the protagonist. They used narrative cloze to evaluate event relatedness, and an order coherence task to evaluate narrative order and reported improvement in both tasks.

In [13], a pattern-based approach is introduced which firstly extracts highly associated verb pairs and their frequency from the web. Then, using co-occurrence data on pairs of verbs, they assessed the strength of the associations by evaluating their mutual information. Finally, using a manually defined threshold they determine whether each association between a verb pair is valid or not. The result is a knowledge base of causal associations of verbs, which contains similarity, strength, antonym, enablement and temporal relations. They did not provide precise evaluation methodology for the obtained results.

Extracting verb-verb, verb-noun and noun-noun event relationship from text [14] concentrated on acquiring causality between events. They used both minimal supervision and unsupervised metrics to learn causal dependencies between two events. They evaluated their model on 20 news articles from CNN. On verbal events, they reported 38.3 % F-score with CEA and 1–2 % improvement using minimally supervised method.

In [21], the authors used Decision Trees for the detection of causations in sentences that contained causal relations. They reported the result of evaluation with precision of 98 % and recall of 84 %, but, their method was not able to detect the causes and the effects.

Using a set of knowledge-rich metrics [17] proposed to learn the likelihood of causal relations between intra- and inter-sentential instances of verb-verb pairs. They relied on the unambiguous discourse markers *because* and *but* to automatically collect training instances of cause and non-cause event pairs, respectively. The result was a knowledge base of causal associations of verbs, which contained three classes of verb pairs: strongly causal, ambiguous and strongly non-causal.

In [18] the authors propose a model for the recognition of causality in intra-sentential verb-noun pairs using Supervised Classifier. They employed linguistic features along with semantic classes of nouns and verbs with high tendency to encode cause or non-cause relations. They generated a test set with instances of form verb-noun phrase and report 46.61 % F-score, and 80.74 accuracy.

In the most recent work, [20] tried to find pairs of verbs linked by a relation explicitly marked by a discourse connector in the corpus, as an indication of a regular semantic relation between the two verbs. The output of this work is the main existing resource that we have compared our results with.

3 DISTRIBUTIONAL MEMORY

Distributional Memory (DM) [11] is a generalized framework for distributional semantics, generalizing different existing typologies of semantic spaces. The aim of Distributional Memory is representing corpus-extracted distributional facts as weighted tuple structures, which are a set of weighted word-link-word tuples $\langle\langle$w1, L, w2$\rangle, \lambda\rangle$. W1 and w2 are a set of strings representing content words, and L is a set of strings representing syntagmatic co-occurrence links between words in a text. Each tuple T has a weight, a real-valued score, assigned by a scoring function $\lambda : W1 \times L \times W2 \rightarrow R$. For example, the tuple $\langle\langle$harvest, before, rain$\rangle$, 66.0141$\rangle$ says *harvest* and *rain* are related through the link *before* with the co-occurrence weight of 66.0141 in the corpus.

DM is built upon the DSM idea. Distributional semantic models (DSMs) are corpus-based models of semantic representation, rely on some version of the distributional hypothesis [24], stating that the degree of semantic similarity between two words (or other linguistic units) can be modelled as a function of the degree of overlap among their linguistic contexts.

Therefore, given a weighted tuple structure, by matricizing the corresponding labelled third-order tensor, four distinct semantic vector spaces can be obtained: $W1 \times LW2$, $W1W2 \times L$, $W1L \times W2$, and $L \times W1W2$. Depending on the tasks, one can choose suitable vector spaces to address it. For instance, one can use $W1 \times LW2$ to tackle attributional similarity tasks such as synonym detection or concept categorization. The $W1W2 \times L$ vectors represent word pairs in a space whose dimensions are links, and can be used to measure relational similarity among different pairs (e.g. $\langle$sergeant, gun$\rangle$ is similar to $\langle$teacher, pen$\rangle$). The $W1L \times W2$ space can be used to capture different verb classes based on the argument alternations they display (e.g. the object slot of kill is more similar to the subject slot of die). The $L \times W1W2$ space displays similarities among links.

Different DM models can be generated based on the selection of the sets W and L and of the scoring function λ. In this paper we used TypeDM, which is the best performing DM model across the various semantic tasks addressed in [11]. The links of TypeDM include lexico-syntactic shallow patterns and, lexicalized dependency paths. Its tensor contains about 130 M non-zero tuples extracted from a corpus of about 2.83 billion tokens. This corpus has been obtained by concatenation of the Web-derived ukWaC corpus, about 1.915 billion tokens, a mid-2009 dump of the English Wikipedia, about 820 million tokens, the British National Corpus, about 95 million tokens. The resulting concatenated corpus was tokenized, POS-tagged and lemmatized with the TreeTagger and dependency-parsed with the MaltParser [30].

The model contains 30 693 lemmas (20 410 nouns, 5 026 verbs and 5 257 adjectives). These terms were selected based on their frequency in the corpus (they are approximately the top 20 000 most frequent nouns and top 5 000 most frequent verbs and adjectives), augmenting the list with lemmas that could be found in various standard test sets, such as the TOEFL and SAT lists.

4 PROPOSED APPROACH

In this section, we introduce our approach to extract semantically related verbs from TypeDM. Figure 1 depicts the structure of our proposed system. Firstly, candidate tuples are extracted from TypeDM. We assume that a verb-noun pair can be a candidate tuple if they are connected through a preposition. Next, tuples that do not contain event pairs are deleted, including

1. tuples containing phrasal verbs,
2. tuples whose w2 are non-action nominals, and
3. tuples whose w2 distinguished as non-action after disambiguation it based on w1 and *link* as context.

Then, after converting action nominals to their corresponding verbs, and aggregating verb pairs, some metrics of relations strength are introduced. Then using *subject* and *object* links in TypeDM, common arguments of semantically related verbs are extracted, which beside common argument weight (CAW), a measure of relation strength, can help to find mapping of verb pairs thematic roles. Finally, we derive the relation direction and relation types including causal, temporal, comparison, and expansion from links connecting two verbs. The following sections describe each of these steps in more details.

4.1 Extraction of Potential Relations

As explained in previous section, the TypeDM tensor contains about 130 M tuples automatically extracted from corpora of about 2.83 billion tokens. In order to get initial tuples that could denote pairs of related events, we have firstly selected 24 links from 25 336 direct and inverse link types formed by syntactic dependencies and patterns. These links are composed of 22 prepositions plus coordination and its inverse

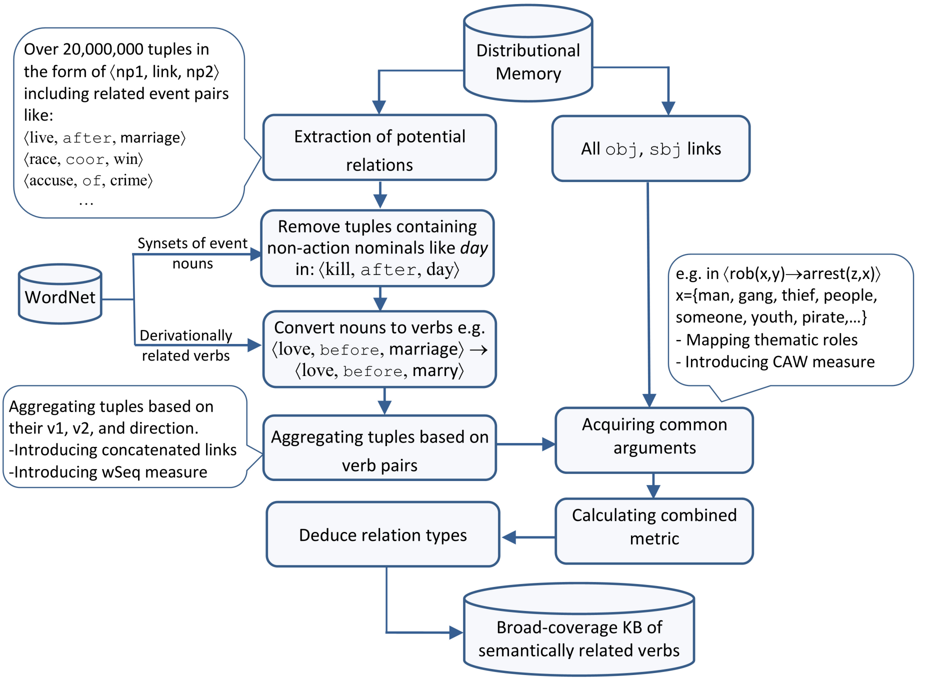

Figure 1. Our proposed system

direction. We extracted all tuples of these 24 links from TypeDM as initial tuples (InitTuples). InitTuples include about 23 M tuples in form of $\langle\langle \mathrm{w1}, \mathrm{L}, \mathrm{w2}\rangle, \lambda\rangle$. In these tuples w1 is mostly a verb (except coordination link) and w2 is always a noun. Table 1 shows these links along with example tuples. For instance, in $\langle$accuse, of, murder$\rangle$, the preposition *of* is a sign of semantic relation between *accuse* and *murder*.

4.1.1 Relation Direction

Semantic relation is an asymmetric relation, so we have to know temporal direction of the relation. That is, in the tuple $\langle$w1, link, w2$\rangle$ we have to know if the relation direction is from w1 to w2 (w1 $\rightarrow$ w2) or from w2 to w1 (w2 $\rightarrow$ w1). In the link set we are working on, the direction of most of links is w2 $\rightarrow$ w1, i.e. w2 happens before w1. The direction of some links, however, is w1 $\rightarrow$ w2. Precisely, the direction of *before* and *coord* are always w1 $\rightarrow$ w2. The direction of all other links except three ambiguous-direction links viz. *for, with*, and *without* is always w2 $\rightarrow$ w1.

We plan to give a solution for finding the relation direction of these three ambiguous links in future. However, at the moment, we simplified the problem and

link	Example tuple	link	Example tuple
after	⟨divorce, after, marriage⟩	since	⟨revise, since, publish⟩
at	⟨win, at, match⟩	through	⟨gain, through, study⟩
because	⟨suffer, because, illness⟩	under	⟨purchase, under, agreement⟩
before	⟨defrost, before, cooking⟩	until	⟨teach, until, retirement⟩
by	⟨learn, by, experiment⟩	upon	⟨renew, upon, expiration⟩
despite	⟨fail, despite, effort⟩	via	⟨melt, via, heating⟩
during	⟨kill, during, raid⟩	while	⟨suffocate, while, feeding⟩
for	⟨marry, for, love⟩	whilst	⟨suspend, whilst, investigation⟩
from	⟨absolve, from, blame⟩	with	⟨charge, with, murder⟩
of	⟨accuse, of, murder⟩	without	⟨capture, without, fight⟩
on	⟨attract, on, offer⟩	coordination	⟨fight, coord , die⟩
over	⟨argue, over, deal⟩	coordination^{-1}	⟨discount, coord-1, price⟩

Table 1. List of links used to extract potentially related event pairs

adapted the majority of directions as the relation direction for these ambiguous links. According to our experiments, more than 95 % of tuples of *with (without)* links have the direction of w2 → w1, so we supposed all tuples in these links (with-without) have direction from w2 to w1. For the link *for*, about 83 % of tuples have the direction of w1 → w2, so we considered its direction as w1 → w2. It should be noted that since each verb pair is usually connected through multiple links (see Section 4.3), their relation direction is introduced by multiple links. Hence the existing error in the direction of *for* and *with* has negligible effect on the direction of final relation.

4.2 Removing Non-Action Nominals

Having extracted InitTuples from TypeDM, the next step is to remove the tuples from it which do not contain event pairs. In natural language, an event is mostly encoded using a verb or a noun. In all tuples $\langle\langle \mathrm{w1}, \mathrm{link}, \mathrm{w2}\rangle, \lambda\rangle$ extracted in previous subsection, w2 is a noun. Obviously, not all these nouns are events or action nominals. Following [25], action nominals are defined as "nouns derived from verbs (verbal nouns) with the general meaning of an action or a process". Also, according to [26], "an event is a situation that occur or happen, and can be expressed by verbs, nominal or some other linguistic units". So, we have to identify action nominals (event nouns) from non-action ones in InitTuples. We have intended to remove three types of tuples that do not contain event pairs, including:

- Tuples where w1 together with a preposition create phrasal verbs like *account for.*
- Tuples where based on WordNet event denoting synsets (WEDS) w2 is not event at all, like *day.*
- Tuples where w2 becomes non-event after disambiguating them based on w1 and preposition. For example *race* is not an event noun in ⟨discriminate, because, race⟩.

After removing these non-event pairs, the number of tuples in InitTuples reduced from over 23 M to about 3.2 M tuples.

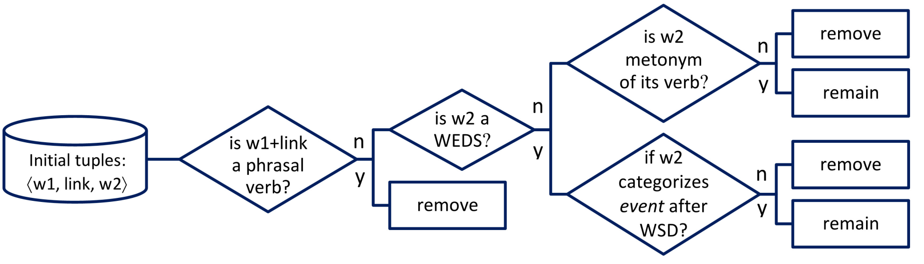

Figure 2. The flowchart of removing tuples containing non-action nominal

4.2.1 Removing Phrasal Verbs

In this article we are interested in prepositions connecting a verb to a noun. On the other hand, in English, prepositions could be combined with verbs producing phrasal verbs like *abide by, accord with, account for, look after*, and so on. This usage of prepositions differs from the one we based our method on. If we let tuples containing phrasal verbs remain in our data, they will produce noise in subsequent steps causing wrong results in relations types' classification and determining relations strength. So, we used a predefined list of phrasal verbs to remove such tuples from InitTuples.

4.2.2 Non-Event Nominals

The term *event* itself has many readings. Some authors use it to refer only to dynamic actions, while others use it to refer also to static situations [42]. In the recent work [38] some definitions of event are provided. The best-known classification of events is one proposed by [41], who distinguishes between *states* (non-dynamic situations persisting over a period of time and without an endpoint, e.g., believe), *activities* (open-ended dynamic processes, e.g., walk), *accomplishments* (processes with a natural endpoint and an intrinsic duration, e.g., build a house) and *achievements* (almost instantaneous events with an endpoint, e.g., find). Moreover, in the linguistic literature, all types of actions, states and processes often fall under the cover term *eventualities*, coined by [32] in his work on the algebra of events. Following Bach's broad notion of event [32], TimeML identifies a wide range of linguistic expressions realizing events, i.e. tensed and untensed verbs (e.g., was captured, to thank), adjectives (e.g., sick), nominals (e.g., strike) and prepositional phrases (e.g., on board).

Investigating various classifications of the event in past literature, we summarize them in Table 2. Respecting these classifications, we can see that three classes exist in almost all classifications viz. state (non-dynamic situations persisting over

Work(authors, year)	Event types	Reference
De Swart, 1998	states, activities, events	[41]
Vendler, 1967	states, activities, event (accomplishments, achievements)	[43]
Bach, 1986	state, process, event (protracted, happenings, culminations)	[34]
Dölling, 2014	point, state, process, event (episode, changes)	[35]
Moens et al, 1988	states, events (culmination, culminated process, point, process)	[36]
de Swart et al, 1999	states, activities, events	[42]
Pustejovsky et al, 2003	state (state, i-state), event (occurrence, aspectual, perception, i-action), reporting	[37]
Pustejovsky, 1991	states, processes, transitions	[38]

Table 2. Event classification in past literature

a period of time and without an endpoint), process (or activity, open-ended dynamic processes), and event (or transition, natural endpoint that are quantized, telic or terminative). Considering the TimeML [35], another class could be added to them i.e. reporting.

On the other hand, like [14, 37] we used WordNet to identify some sub-trees from WordNet synsets so that their hyponym (children) are mostly action nominals. Surprisingly, the top four WordNet synsets which we have gained with highest ratio of event nominals are analogous to the four classes that we acquired from the previous literature. These synsets along with some of their hyponym (children) nouns are shown in Table 3. Indeed, not all nouns under these synsets are action nominals. However, as our first goal is identifying and discarding tuples containing non-event nouns, this method works well at present.

Synsets	Sample event nouns
event	work, revolution, death, service, act, birth, flight, game, development, war, crime
process	pressure, loss, review, implication, access, growth, investigation, consideration, assessment
state	operation, fear, interest, need, love, injury, marriage, press, absence, independence
message	comment, note, notice, agreement, request, advice, statement, claim, regulation, instruction
symptom	pain, suffering, scar, founder, fever, congestion, burn, inflammation, tenderness, rash, cough

Table 3. WordNet event denoting synsets and their equivalent classes in the past literature

Given WEDS, we can determine if a noun have the chance of being event noun or not. For instance, the leaf-to-root paths for the first sense of offense(n) and album(n) are as follow, respectively:

- offense ⇒ behavior ⇒ activity ⇒ act ⇒ **event** ⇒ psychological feature ⇒ abstraction ⇒ entity,
- album ⇒ medium ⇒ instrumentality ⇒ artifact ⇒ whole ⇒ object ⇒ physical entity ⇒ entity.

The *event* synset in leaf-to-root hypernym path of the first sense of *offense* indicates that this noun could be an action nominal. For the word *album*, on the other hand, there is not such synset, neither in its first sense nor any other senses.

This implies that *album* could not denote an event. Using this method, we determine the sense number of the nouns that can denote event together with the synsets name. We call this information semantic-category. In order to determine semantic-category for a noun, the leaf-to-root hypernym path for its all senses is searched. That is, we have gone through all its WordNet senses; have examined their hypernyms (parent) in WordNet hypernym relations one-by-one upward. During the search, for each sense S of the noun, if one of the WEDS synset is found, we assigned the synset name along with the sense number of S, otherwise its value will be non-action.

For example *event/4* for semantic-category of a noun means that hypernym path of its fourth sense contains *event*, and *process/1* means that hypernym path of its first sense contains *process*, and so forth. For example semantic-category for *birth* and *authority* are:

- semantic-Cat (birth) = $\langle$event/2 $-$ process/3$\rangle$,
- semantic-Cat (authority) = $\langle$state/4$\rangle$.

Algorithm 1 shows the details for extracting semantic-cat for a word. Firstly, the semantic cat is set to an empty string. Then, by iterating through all noun senses of the word and checking their hypernym against event denoting synsets, semantic-cat is acquired.

Algorithm 1 Semantic-cat extracting algorithm

```
Input: word
Output: semantic categories
semantCat ← empty
nounSenses ← all noun senses (word)
  for S∈ nounSenses do
    HoS ← all hypernyms of S
      for h∈ HoS do
         If h is in {event, process, state, message, symptom} then
           append(semantCat, h+"/"+sense_number(S))
         end if
      end for
  end for
If semantCat is empty then
  semantCat ← non-event
end if
return semantCat
```

4.2.3 Metonym Nouns

Beside action nominals that can be identified through Algorithm 1, there are other nouns that are of our interest. Although semantically can denote event, these nouns are categorized as non-event nouns by Algorithm 1. Considering the tuple $\langle \text{escape}, \text{from}, \text{jail} \rangle$ for example, we can understand that event *jail* can result in event *escape*. In fact, the noun *jail* in this tuple can denote event *jail* (putting in jail). However, *jail* is a noun in this tuple and semantic-Cat (jail) is non-event. The point is that the word *jail* has a verb form as well, which is an event. Actually, here the noun *jail* can be metonym of its verb in our method. Another such example is $\langle \text{receive}, \text{after}, \text{pay} \rangle$, where noun *pay* denotes event of paying, while it is categorized as non-event nouns by Algorithm 1 as well. There are many such nouns in TypeDM which are categorized as non-event while could denote an event. In order to detect these metonym nouns, we have heuristically chosen nouns having two following criteria:

- the noun has a verb form with the same spelling,
- the noun categorizes as non-event based on Algorithm 1.

We found about 800 such nouns in TypeDM through above-mentioned criteria. Table 4 shows some examples of such nouns extracted from TypeDM.

w1	link	w2
remove	through	filter
cut	with	saw
deduct	under	pay
grow	because	feed
run	on	steam
increase	despite	drop
access	via	join
defrost	before	cook
escape	by	hide
win	after	elect
charge	for	fuel

Table 4. Some examples of tuples where w2 is a noun that can be metonym of its verb

4.2.4 Disambiguate Polysemous Words

Many polysemous words in English can have both event and non-event meanings. For instance the word *spring* can be both non-event noun (springtime, fountain, a metal elastic device, and elasticity) and event noun (leap). We have to identify and separate non-event nouns like *spring* in $\langle \text{occur}, \text{during}, \text{spring} \rangle$ or *race* in $\langle \text{discriminate}, \text{because}, \text{race} \rangle$ to prevent probable harmful side-effects in subsequent steps. We tried to convert such triples to a sentence and use state of the art WSD like BabelNet [27] to find correct sense of the ambiguous noun. Regarding this approach there are two points. Firstly, it is not easy to convert every tuple to a well formed English sentence to fit input of WSD like BabelNet. Secondly, the precision of this approach is very low in our data. For example, BabelNet disambiguate *race* in tuple $\langle \text{discriminate}, \text{because}, \text{race} \rangle$ as *any competition* when we tried it in the sentence *"People should not be discriminated because race"*. So we decided to build a model for disambiguating w2 in our tuples.

To do so, we have to construct a model which, given a tuple $\langle \mathrm{w1}, \mathrm{preposition}, \mathrm{w2} \rangle$, will correctly predict the sense to which the w2 belongs using w1 and preposition as context. This is a classification problem, which for a w2 with N noun senses, has N different classes. For each of these N senses, we extracted salient words from the WordNet glosses, synonyms, and hypernyms as feature set. Also, in order to convert the context (i.e. the w1 and the preposition) to a set of feature words, we used TypeDM. Specifically, for a tuple $\langle \mathrm{w1}, \mathrm{preposition}, \mathrm{w2} \rangle$ in which w2 is an ambiguous noun, we extracted all tuples $\langle \langle \mathrm{w1}, \mathrm{L}, \mathrm{w2} \rangle, \lambda \rangle$ having the pattern $\langle \mathrm{w1}, \mathrm{preposition}, * \rangle$ from TypeDM. Then, we have chosen top 5 w2 of the extracted tuples having highest λ value. For example, in tuple $\langle \mathrm{occur}, \mathrm{during}, \mathrm{spring} \rangle$ it is unknown for the system if the *spring* means a season, outflow, a metal, or leap. After extracting top nouns for the pattern $\langle \mathrm{occur}, \mathrm{during}, * \rangle$ we came up with following nouns: *season, phase, summer, winter, stage*. Using these nouns as context, and comparing similarity between it and the features of each senses of the noun *spring*, we got spring #1 as most similar sense, which means the season of growth. Applying this idea on tuple $\langle \mathrm{pump}, \mathrm{from}, \mathrm{spring} \rangle$ gives following nouns as context: *mine, pit, stream, station*, and *source*. After calculating similarities between all *spring-n* senses and the context words, we got the spring #2 as the result which means a natural flow of ground water. We evaluated this WSD method on 200 tuples of InitTuples containing ambiguous nouns as their w2 as test data. The correct senses of ambiguous nouns in these tuples were identified by two human annotators. We have achieved a 0.78 kappa score for the human inter-annotator agreement. Evaluating the WSD on this test data yields the accuracy of 74 %.

4.2.5 Mapping Semantic-Cat To a Real Number

In the Subsection 4.2.2, we used semantic-cat to remove tuples containing non-action nominals. However, it can also be used to rank action nominals, based on how likely they can refer to an event. In other words, nouns under some WEDS synsets refer to event more often than some others. For example synsets like *event, process* are more event-denoting than *state*. Additionally, sense numbers of the nouns that belong to these synsets are important as well, e.g., *event/1* is more probable to denote an event than *event/2*, and it is relative to *event/3*, and so on. In order to capture action denoting strength of any action nominals, we converted semantic-cat to a real numbers called *catVal*. To obtain the value of catVal we used two metrics:

1. the ratio of the nouns belong to the synset that denote event,
2. the sense number of the noun that denotes event.

The less the sense number is, the higher is the catVal value. Algorithm 1 (italic lines) shows details of this calculation. CatVal will be used in ranking step in Section 4.5.

4.3 Aggregating Tuples Based on Verb Pairs and Direction

As explained in Section 3, there is a co-occurrence frequency of the tuple (λ) to characterize its statistical salience; however, it is not accurate enough to determine the real strength of the semantic relations solely. On the other hand, a pair of events may be related by different links in different tuples. So, we decided to aggregate the tuples based on their event pairs and the relation direction. The event pairs in *InitTuples* are now in the form of verb-noun pairs. Although some of these nouns may denote the same event, however, they may have different derivational forms like *graduation, graduating, graduate*, etc. So, we decided to convert w2 nouns to their corresponding verbs to get verb-verb pairs. In addition to solving the problem of tuple aggregation, this conversion is also necessary to find common arguments in the next section. We used derivationally related form API of WordNet to convert nouns to their corresponding verb(s), see Figure 3 b).

Having verb-verb pairs in the tuples, we now can aggregate the tuples based on their verb pairs and the relation direction, summing the co-occurrence frequency and concatenating the links for them. This way, verb pairs that are related through different links (in different tuples) will be connected through concatenation of that links (converted to just one tuple). This grouping process reduces the number of tuples (distinct verb pairs) in our KB to about 1.5 M tuples. Doing so, the weight of each verb pair is now the sum of the weights of all tuples that have been grouped to create that pair. This new sum is more accurate to capture the strength of semantic relationship between two verbs. We call this new metric *wSequence* or *wSeq* for short. Figure 3 shows this process for the verb pair *admit-graduate.*

(a) Verb-noun tuples for the pair admit-graduate.

W1	link	W2	dir	λ
admit	upon	graduation	$\leftarrow$	24.98
admit	after	graduation	$\leftarrow$	21.00
admit	coord	graduation	$\rightarrow$	19.49
admit	after	graduating	$\leftarrow$	7.01
admit	with	graduating	$\leftarrow$	4.88
admit	for	graduate	$\rightarrow$	2.15
admit	of	graduate	$\leftarrow$	1.41

1

(b) Verb-verb tuples for the pair admit-graduate.

W1	link	W2	dir	λ
admit	after	graduate	$\leftarrow$	28.01
admit	upon	graduate	$\leftarrow$	24.98
admit	coord	graduate	$\rightarrow$	19.49
admit	with	graduate	$\leftarrow$	4.88
admit	for	graduate	$\rightarrow$	2.15
admit	of	graduate	$\leftarrow$	1.41

2

(c) Aggregate the tuples based on verb pair and the direction for the pair admit-graduate.

W1	concatenated links	W2	dir	wSeq
admit	after (28.01), upon (24.98), with (4.88), of (1.41)	graduate	$\leftarrow$	59.28
admit	coord (19.49), for (2.15)	graduate	$\rightarrow$	21.64

Figure 3. a) Initial tuples of pair admit-graduate, b) convert nouns to their corresponding verbs, c) aggregate the tuples based on verb pairs and the direction

4.4 Roles of Common Arguments

Considering the fact that semantically related verbs should have common arguments, we believe that the more two verbs are semantically related, the more words they will have as their common arguments (subject or object). For instance, *plant* and *harvest* which are semantically related have many words that can be their common arguments, but *plant* and *crash*, which are not semantically related, have almost no word as their common arguments:

- Argument (plant) $\cap$ Argument (harvest) $=$ $\{\text{crop}, \text{plant}, \text{grape}, \text{seed}, \text{potato}, \text{grain}, \text{fruit}, \text{corn}, \text{wheat}\}$,
- Argument (plant) $\cap$ Argument (crash) $= \emptyset$.

We call these words that can be arguments of both verbs *common arguments*. Common arguments can be found in subject and object links in TypeDM. There are more than 10 M such links in TypeDM. We define Common Argument Weight (CAW) as the relative measure of the strength between two verbs. To acquire this value for two given verbs *verb1* and *verb2*, we have firstly chosen subject and object links (tuples) of TypeDM for them, namely V1Links and V2Links, respectively. There are about a few thousands such links for each verb in TypeDM. Then we joined the tuples of V1Links and V2Links based on their common arguments to get joint tuples *jointTuples*. That is, for each tuple of V1Links $\times$ V2Links if V1Links.arg equals V2Links.arg, we keep (join) them, otherwise discard them. Then, we calculate $f(\lambda 1, \lambda 2)$ as a function of $\lambda 1$ and $\lambda 2$ of the joint tuple, where $\lambda 1$ is the weight of verb1 tuple and $\lambda 2$ is the weight of verb2 tuple. Lastly, by sorting the joint tuples based on $f(\lambda 1, \lambda 2)$ and picking the highest one, CAW can be calculated as a function of $\lambda 1, \lambda 1$ of this tuple. See Algorithm 2 for more details.

Algorithm 2 has three outputs, CAW, common arguments and role mapping. Common arguments can be acquired by selecting common arguments of top n tuples from sortedTuples, sorted tuples of jointTuples based on $f(\lambda 1, \lambda 2)$. Role mapping maps the thematic roles of related verbs (e.g., the Agent of kill is mapped to the Patient of arrest). This is very useful information about semantically related verbs that can be used in many NLP applications, like Coreference Resolution. In order to get this mapping, we have heuristically chosen the rel1 and rel2 of the top 1 tuple from sortedTuples. Although this is only a heuristic, but in most cases it works properly. The rationale behind it is that the common argument that comes with both verbs most of the times has a certain role with each verb. Hence, choosing the top 1 tuple of the sortedTuples which has the highest value of λc is a simple and acceptable solution for this problem. The verb pair (escape, arrest), for instance, has nouns like *prisoner, criminal, man* as their common arguments which are usually subject of *escape* and object of *arrest*. So, for this verb pair, we obtain the mapping escape (sbj) = arrest (obj).

Although gathered from big parsed corpora and not necessarily co-occurring in the same document, the acquired common arguments are so accurate. Beside

Algorithm 2 Algorithm for extraction of CAW, Role Mapping, and Common Arguments

```
Input: verb1, verb2
Outputs: sbj, obj mapping, CAW, common arguments
V1Links ← all the links of TypeDM where w1= verb1 and link ∈ {sbj, obj}
V2Links ← all the links of TypeDM where w1= verb2 and link ∈ {sbj, obj}
    jointTuples ← empty
    for t1∈ V1Links do
      for t2∈ V2Links do
         if t1.arg=t2.arg then
              CA ← t1.arg
              λ1 ← t1.λ
              λ2 ← t2.λ
              rel1 ← t1.rel
              rel2 ← t2.rel
              λc ← f(λ1, λ2)
              jointTuples.add (CA, λ1, λ2, λc, rel1, rel2)
         end if
    end for
  end for
  sortedTuples ← jointTuples.sortDescending(λc)
  CAW ← π_λc(σ_top1 sortedTuples)
  RoleMapping ← π_rel1, rel2(σ_top1 sortedTuples)
  commonArgs ← π_CA(σ_top n sortedTuples)
```

a metric for relations strength measurement, common arguments can act as a mean to disambiguate polysemous verb with respect to another verb. For instance in (install, execute) common arguments are words denoting a program or script, which indicates *execute* means run a program, but in (arrest, execute) common arguments denote a prisoner or criminal, which indicates *execute* means put to death. Table 5 shows some examples of CAW, common arguments and role mapping.

Verb pair	Common arguments	Role mapping	CAW
design- print	page, poster, card, form, leaflet, book, work, logo, character, map, stamp, piece, line, part, cover, section, t-shirt, pattern	obj-obj	685.7961
sow- harvest	crop, seed, field, grain, plant, corn, wheat, bean, barley, onion, variety, vegetable, rice, flower, carrot	obj-obj	651.5708
rob-arrest	man, gang, thief, people, someone, youth, pirate, bandit, government, criminal, robber, soldier, guy, thug, group, woman, burglar, gunman	sbj-obj	258.8246
Try-succeed	government, man, company, party, student, team, child	sbj- sbj	396.6854
own-manage	Business, company, property, site, estate, asset, land, team, farm	obj-obj	1227.6482

Table 5. Some examples of CAW, common arguments and role mapping

4.4.1 Calculating $f(\lambda 1, \lambda 2)$

In above subsection we expressed $f(\lambda 1, \lambda 2)$ as a function that combines $\lambda 1$ and $\lambda 1$ as a single metric which we called λc. We decided to choose the minimum of $\lambda 1$ and $\lambda 2$ as $f(\lambda 1, \lambda 2)$, i.e. $f(\lambda 1, \lambda 2) = \text{minimum}(\lambda 1, \lambda 2)$. One may wonder why we have chosen minimum not maximum or multiplication of $\lambda 1$ and $\lambda 2$, for example. We opted for the minimum for two reasons. First, it is obvious that both of $\lambda 1$ and $\lambda 2$ are important in weighting the joint tuple of v1Link and v2Link, so we have to use a function of both values. Second, if we choose multiplication or maximum or average of $\lambda 1$ and $\lambda 2$, it may cause undesirable results, because the common argument may come with verb1 (verb2) more often than the other, resulting in a big value for $\lambda 1(\lambda 2)$. So if we multiply, sum, or choose the maximum value of $\lambda 1$ and $\lambda 2$, we will get a high value of CAW for a verb pair that may not agree with their common arguments.

4.5 Calculating Combined Metric

Since the beginning of Section 4, we introduced three metrics that can be used in ranking semantically related verbs based on the relations strength, i.e., catVal, wSeq, and CAW. In this section, after introducing a new metric, we plan to combine them to obtain combined metric.

In addition to the metrics introduced in previous subsections, we can use PMI (Pointwise Mutual Information). PMI (Equation (1)) is information-theory approach to measure the statistical association between two words. In our dataset, PMI estimates whether the co-occurrence of two verbs is higher than the a priori probability of them occurring independently. PMI defined as:

$$PMI(v1, v2) = \log\left(\frac{P(v1, v2)}{P(v1)P(v2)}\right). \tag{1}$$

The value of $P(v1, v2)$ can be obtained from the of co-occurrence weight of two words acquired from the corpus (Section 3). For calculating the value of $P(v1)$, the sum of λ in all tuples T where $v1 \in T$ is computed. $P(v2)$ is calculated in a similar way.

Now there are four metrics which can be used to rank tuples of verb pairs based on their relation strength. In order to create the combined ranking formula based on these metrics, we ranked 150 verb pairs manually and used them as train data of a linear regression model.

4.6 Deduce Relation Types

To classify semantic relations, following Penn Discourse Treebank (PDTB) [28], we grouped discourse relations into four classes: causal relations (Contingency), temporal relations (Precedence, Succession), comparison relations (Contrast), and expansion relations (Conjunction).

Contingency is used when the connective indicates that one of the events causally influences the other.

Temporal is used when the connective indicates that the situations described in the arguments are related temporally.

Comparison applies when the connective indicates that the relation highlight prominent differences between the two situations.

Expansion covers those relations which expand the discourse by providing additional information or illustrating alternative situations.

To acquire relation types between verb pairs, we used their connecting links. As explained in previous subsections, there are 24 links connecting verb pairs in our KB (see Table 1). After aggregating tuples based on verb pairs and the relation direction, every verb pair is connected through a subset of these links (Figure 3). Now the problem is to infer relation type(s) from the connecting links. For each relation, there are some linguistic cues to infer it from Table 6 shows these cue links.

Our introduced cue words for each relation are in accordance with the abovementioned definitions of those relations. In temporal relation, for example, the definition says "the connective indicates that the situations described in the arguments are related temporally". Each of our introduced cue words for temporal relation is such indicative without no exception or ambiguity. The *after* and *before* prepositions denote *succession* and *precedence* relations, respectively, which are subtypes of the Asynchronous temporal relation introduced in PDTB. *Until* and *upon* denote succession relation as well. The prepositions *at, on, while, during, whilst*, and *over* denote Synchronous relations which is subtype temporal relation introduced in PDTB. Some of these prepositions could denote other meanings than those of temporal relation. For example, *at* and *on* could denote position or location, but as we removed non-event arguments for these prepositions in Section 4.2, they will just denote temporal relation in existing tuples. For the comparison (contrast) and expansion (Conjunction) relations, the selected cue words in Table 6 are in accordance with PDTB definitions for these relations.

Relation type	Cue word links
causal	through, by, via, with, of, for, from, because, since, under
temporal	after, at, on, over, before, until, upon, while, during, whilst
comparison	despite, without
expansion	coord , coord-1

Table 6. Cue word links for each relation type

For three relations, i.e. *temporal*, *comparison*, and *expansion* we can use a rule, based on their cue words to infer the existence of that relation between each verb pair. That is, these three relations can be identified by this simple rule: for each relation $R \in \{\text{temporal}, \text{comparison}, \text{expansion}\}$ if the words in connecting link between verbs $v1$ and $v2$ contain a cue word of the relation R, then the relation R holds between $v1$ and $v2$.

For the *cause*, on the other hand, we could not find any rule that works based on its introduced cue words. For instance, in the tuple ⟨charge, with, offend⟩ the preposition *with* is a sign of causal relation between *offend* and *charge*, but in ⟨answer, with, laugh⟩ there is no causal relation between *laugh* and *answer*. The problem is that these connectors are ambiguous in that they are associated with several relations. In addition, the amount of links contribution in relations (i.e. the co-occurrence weight of the tuple in corpus) is not 1, 0 modes (i.e. exist or not exist) but they can take a value ranging from small amount to several hundreds. We believe that the links values are also important in determining the relations. Hence, sometimes three cue words with relatively low co-occurrence value (through (10), by (5), via (2)) could denote the cause relation and sometimes just two cue words with high co-occurrence value (with (200), by (170)) is enough. Sometimes one cue word like *because* can solely be translated to the cause relation. These all indicate that generalizing the links to get a rule to translate links set to the *cause* relation is not easy. So we decided to use a learning method for this task.

In order to acquire training data, we manually collected instances of cause and non-cause event pairs from the KB. We labeled 500 cause and 500 non-cause verb pairs. We used the value of connecting links as input features to train a supervised model for classifying relation between verb pairs as cause or non-cause. We chose the Random Forest classifier implemented in Weka [29] to train the model which yielded the highest performance.

5 EVALUATION AND DISCUSSION

In this section, we present the evaluation of our semantically related verbs KB. Specifically we performed experiments to evaluate

(1) the direction between verbs pairs,

(2) the ranking of verb pairs based on their strength of association,

(3) the quality of the four categories relations between verb pairs in KB (i.e., causal, temporal, comparison, and expansion),

(4) the mapping between thematic roles of verb pairs.

For each experiment we created a test data from our KB and asked human annotators to annotate them. Also, for cause relation of case (3) which is most important semantic relation in our task, we compared the performance of our approach with knowledge bases that are extracted in similar way. This experiment is done against available data sets of causal relations that are explained in following subsection. We consulted the freely available resources VerbOcean [13] and V2R [20]. VerbOcean data contains 98 362 tuples including 58 330 distinct verb pairs. V2R contains over 8 000 000 tuples including 3 803 294 distinct verb pairs. It should be noted that the tuples in V2R contain some prefixes or affixes that could affect normal comparison.

For example many tuples contain a [not] or [state verb] prefix. We removed these affixes before comparison. We also removed its tuples which contained non-word tokens (e.g. numbers, quotation, exclamation mark).

5.1 Data Sets

5.1.1 Available Data

This section presents details of freely available data for cause-effect relations. The details of this data set are explained below.

SemEval-2: SemEval-2 Task 8 focuses on multi-way classification of semantic relations between pairs of nominals. One of these relations is Cause-Effect (CE). In the original dataset in each sentence one causal pair has been annotated. We have extracted these pairs. Because the events in VerbOcean, V2R and our KB are expressed through verb pairs, we converted event pairs extracted from SemEval to their corresponding verbs (if possible). This way we obtained 451 Cause-Effect verb pairs.

WordNet cause relations: WordNet contains causal relations between verb pairs. Extracting these relations from WordNet we obtained 743 cause verb pairs.

ECED: in [14] the authors have annotated some causal relations from news documents and used the data in developing and evaluation of their method. We extracted causal event pairs from these annotated data and converted event nouns to their corresponding verbs. This way we get 400 cause-effect verb pairs. We called these data ECED.

As the first experiment, we compared our method with V2R and VerbOcean against above data. This experiment tests the coverage of causal relation along with the direction of relation in our KB. Figure 4 and Table 7 show the results.

The better coverage of our result in comparison with V2R becomes more valuable when taking this point into account that the total number of **distinct** verb pairs in our KB is far lesser that of in V2R (0.9 M vs. 3.8 M). This means that besides the coverage, the precision of our method is much higher.

Data	Methods	Total relations	Included verb pairs	recall
WordNet cause relation	Our method	743	**256**	**35.45%**
	V2R	743	218	29.34%
	VerbOcean	743	16	2.15%
Do et al	Our method	400	**278**	**69.50%**
	V2R	400	232	58.00%
	VerbOcean	400	29	7.25%
SemEval-2010	Our method	451	**114**	**25.27%**
	V2R	451	82	18.18%
	VerbOcean	451	6	1.33%

Table 7. Coverage of cause relation in our KB, V2R, and VerbOcean with respect to ECED, WordNet cause relation and SemEval-2010

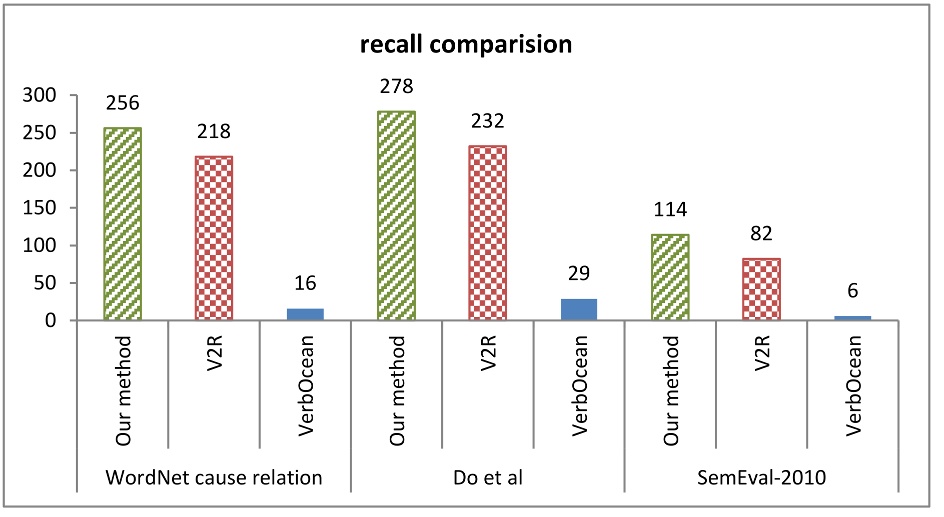

Figure 4. Recall comparison of our method against V2R and VerbOcean

5.1.2 Annotated Data from Our KB

We collected test data from our KB for experiments (1) to (4). These test sets were selected randomly with equal proportion of weak and strong relation strength. We asked two annotators to annotate these data. Then we tested our KB against these test data. The process is explained in more details below.

We selected 100 verb pairs from our KB randomly to create two test sets for experiments (1) and (4), respectively. Then we asked two annotators to identify the direction of relations between each pair (experiment 1) and find the mapping between thematic roles of the verb pairs (experiment 4). The kappa score for the human inter-annotator agreement achieved on Test-set1 (Test-set4) is 0.94 (0.51), respectively. Then we compared the direction of these tuples with that of our KB. The precision for relation direction and thematic role mapping was 91 % and 46 %, respectively.

For experiment (3), i.e. evaluating the quality of the four categories relations between verb pairs in KB, we created a test data from our KB. For this purpose, we selected 100 verb pairs for each of four categories randomly. These data were annotated by two human annotators to determine if the semantic relation holds between the verb pairs of each test set or not. They were provided with annotation guidelines where it was needed. For instance, the cause relation is hard to identify, so we adopted the annotation guidelines from [31, 18] which are as follows: *"Assign cause label to a pair* (a, b)*, if the following two conditions are satisfied: (1) a temporally precedes/ overlap in time, (2) while keeping as many state of affairs constant* *as possible, modifying a must entail predictably modifying b. Otherwise assign non-cause label"*. The kappa scores for the human inter-annotator agreement achieved on causal, temporal, comparison, and expansion Test-sets are 0.51, 0.91, 0.74, and 0.62, respectively. We compared our KB against these data. Table 8 shows the results.

Relation type	precision
causal	37.42%
temporal	75.69%
comparison	62.33%
Expansion	50.26%

Table 8. Precision of our relations

For experiment (4), i.e. the task of ranking verb pairs based on the strength of relations, we randomly selected 10 verbs. Two annotators were asked to sort related verbs of each 10 verbs based on the strength of their association. We employed Spearman's rank correlation co-efficient (Equation (2)) to compare the ranked list of verb pairs based on the scores of our metrics and the rank given by the human annotators.

$$P = \frac{n(\sum x_i y_i) - (\sum x_i)(\sum y_i)}{\sqrt{n(\sum x_i^2) - (\sum x_i)^2}\sqrt{n(\sum y_i^2) - (\sum y_i)^2}}. \tag{2}$$

Here, n is the total number of verb pairs in the test set, x_i is the human annotation rank and y_i is the metric's rank of verb pairs of the test set. Spearman's rank correlation coefficient has a range of $[-1, 1]$. A coefficient of -1 corresponds to the two lists being perfectly negatively correlated (one is the reverse sort of the other), a coefficient of 1 corresponds to perfect correlation, and a coefficient of 0 for rankings being completely independent.

Table 9 shows the results of evaluating introduced metrics for relation strength. The A, B, C, D, and E schemas are λ, wSeq, wSeq + CAW, wSeq + CAW + catVal, and combined metrics, respectively.

Metric schema	A	B	C	D	E
P score	0.2561	0.34993	0.57852	0.61254	0.74125

Table 9. The Spearman's rank correlation coefficient for the metrics

6 CONCLUSION

Providing a repository of semantic relation between verbs is of great importance in various NLP applications including Question Answering [1, 2], Machine Translation, Information Extraction, Coreference Resolution [3], Prediction [4], Summarization [5], Recognizing Textual Entailment [6], etc.

In this paper, we discussed how parsed data of a big corpus could have a significant impact on creating semantic relations repository between verbs. We used both verb and action nominals as event triggers. Incorporating connecting links between event pairs, we tried to classify relations types to categories such as causal relations, temporal relations, comparison relations, and expansion relations. In order to determine the strength of association between verb pairs, we introduced some numerical measures including wSeq, CAW, catVal, and PMI. We evaluated our work against two freely available resources of semantic verbs. As reported in the evaluation section, the result was promising.

On the other hand, one limitation in our work is that there is no phrasal verb in our repository. The reason is that the parser used in parsing source corpora did not distinguish between particle and preposition. In other word, it treats *put on shoulder* and *put on clothes* the same, while in the former *on* is a preposition and in the latter it is a particle. This way, for a sentence like *put on clothes* we wrongly have *put* as the verb. This has a negative effect on the quality of our results.

REFERENCES

[1] Girju, R.: Automatic Detection of Causal Relations for Question Answering. Proceedings of the ACL 2003 Workshop on Multilingual Summarization and Question Answering (MultiSumQA '03), Vol. 12, 2003, pp. 76–83, doi: 10.3115/1119312.1119322.

[2] Higashinaka, R.—Isozaki, H.: Automatically Acquiring Causal Expression Patterns from Relation-Annotated Corpora to Improve Question Answering for Why-Questions. ACM Transactions on Asian Language Information Processing, Vol. 7, 2008, No. 2, Art. No. 6, doi: 10.1145/1362782.1362785.

[3] Bean, D.—Riloff, E.: Unsupervised Learning of Contextual Role Knowledge for Coreference Resolution. Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2004), 2004, pp. 297–304.

[4] Radinsky, K.—Horvitz, E.: Mining the Web to Predict Future Events. Proceedings of the Sixth ACM International Conference on Web Search and Data Mining (WSDM '13), 2013, pp. 255–264, doi: 10.1145/2433396.2433431.

[5] Marcu, D.: From Discourse Structures to Text Summaries. Proceedings of the ACL, 1997, pp. 82–88.

[6] Pazienza, M. T.—Pennacchiotti, M.—Zanzotto, F. M.: Mixing Wordnet, Verbnet and Propbank for Studying Verb Relations. Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC '06), 2006, pp. 1372–1377.

[7] Baker, C. F.—Fillmore, C. J.—Lowe, J. B.: The Berkeley FrameNet Project. Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics –

Volume 1 (ACL '98/COLING '98), Montreal, Canada, 1998, pp. 86–90, doi: 10.3115/980845.980860.

[8] Kipper, K.—Dang, H. T.—Palmer, M.: Class-Based Construction of a Verb Lexicon. Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence (AAAI/IAAI), 2000, pp. 691–696.

[9] Kingsbury, P.—Palmer, M.—Marcus, M.: Adding Semantic Annotation to the Penn Treebank. Proceedings of the Human Language Technology Conference, 2002, pp. 252–256.

[10] Miller, G. A.: WordNet: A Lexical Database for English. Communications of the ACM, Vol. 38, 1995, No. 11, pp. 39–41, doi: 10.1145/219717.219748.

[11] Baroni, M.—Lenci, A.: Distributional Memory: A General Framework for Corpus-Based Semantics. Computational Linguistics, Vol. 36, 2010, No. 4, pp. 673–721, doi: 10.1162/coli_a_00016.

[12] Chang, D.-S.—Choi, K.-S.: Causal Relation Extraction Using Cue Phrase and Lexical Pair Probabilities. In: Su, K. Y., Tsujii, J., Lee, J. H., Kwong, O. Y. (Eds.): Natural Language Processing – IJCNLP 2004. Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 3248, 2004, pp. 61–70, doi: 10.1007/978-3-540-30211-7_7.

[13] Chklovski, T.—Pante, P.: VerbOcean: Mining the Web for Fine-Grained Semantic Verb Relations. Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP), Vol. 4, 2004, pp. 33–40.

[14] Do, Q. X.—Chan, Y. S.—Roth, D.: Minimally Supervised Event Causality Identification. Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '11), 2011, pp. 294–303.

[15] Kozareva, Z.: Cause-Effect Relation Learning. Workshop Proceedings of TextGraphs-7 on Graph-Based Methods for Natural Language Processing (TextGraphs-7 '12), 2012, pp. 39–43.

[16] Mirza, P.—Tonelli, S.: An Analysis of Causality Between Events and Its Relation to Temporal Information. Proceedings of COLING 2014, the 25$^{\text{th}}$ International Conference on Computational Linguistics: Technical Papers, 2014, pp. 2097–2106.

[17] Riaz, M.—Girju, R.: Toward a Better Understanding of Causality Between Verbal Events: Extraction and Analysis of the Causal Power of Verb-Verb Associations. Proceedings of the Annual SIGDIAL Meeting on Discourse and Dialogue (SIGDIAL 2013), 2013, pp. 21–30.

[18] Riaz, M.—Girju, R.: Recognizing Causality in Verb-Noun Pairs via Noun and Verb Semantics. Proceedings of the EACL 2014 Workshop on Computational Approaches to Causality in Language (CAtoCL), 2014, pp. 48–57, doi: 10.3115/v1/W14-0707.

[19] Riaz, M.—Girju, R.: In-Depth Exploitation of Noun and Verb Semantics to Identify Causation in Verb-Noun Pairs. In 15$^{\text{th}}$ Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL), 2014, pp. 161–170, doi: 10.3115/v1/w14-4322.

[20] Conrath, J.—Afantenos, S.—Asher, N.—Muller, P.: Unsupervised Extraction of Semantic Relations Using Discourse Cues. Proceedings of COLING 2014, the

25th International Conference on Computational Linguistics: Technical Papers, 2014, pp. 2184–2194.

[21] Sorgente, A.—Vettigli, G.—Mele, F.: Automatic Extraction of Cause-Effect Relations in Natural Language Text. In: Lai, C., Semeraro, G., Giuliani, A. (Eds.): Proceedings of the 7th International Workshop on Information Filtering and Retrieval. CEUR Workshop Proceedings, 2013, Vol. 1109, pp. 37–48.

[22] Chambers, N.—Jurafsky, D.: Unsupervised Learning of Narrative Schemas and Their Participants. Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP (ACL '09), 2009, pp. 602–610, doi: 10.3115/1690219.1690231.

[23] Chambers, N.—Jurafsky, D.: Unsupervised Learning of Narrative Event Chains. Proceedings of ACL-08: HLT, 2008, pp. 789–797.

[24] Harris, Z. S.: Distributional Structure. Word, Vol. 10, 1954, No. 2-3, pp. 146–162, doi: 10.1080/00437956.1954.11659520.

[25] Comrie, B.: The Syntax of Action Nominals: A Cross-Language Study. Lingua, Vol. 40, 1976, No. 2–3, pp. 177–201, doi: 10.1016/0024-3841(76)90093-0.

[26] Pustejovsky, J.—Hanks, P.—Sauri, R. et al.: The TIMEBANK Corpus. Proceedings of Corpus Linguistics, 2003, pp. 647–656.

[27] Moro, A.—Cecconi, F.—Navigli, R.: Multilingual Word Sense Disambiguation and Entity Linking for Everybody. Proceedings of the 2014 International Semantic Web Conference – Posters and Demonstrations Track (ISWC-PD '14). CEUR Workshop Proceedings, Vol. 1272, 2014, pp. 25–28.

[28] Prasad, A. J.—Miltsakaki, E.—Dinesh, N.—Lee, A. et al.: The Penn Discourse Treebank 2.0 Annotation Manual. The PDTB Research Group, 2007.

[29] Hall, M.—Frank, E.—Holmes, G.—Pfahringer, B.—Reutemann, P.—Witten, I. H.: The WEKA Data Mining Software. ACM SIGKDD Explorations Newsletter, Vol. 11, 2009, No. 1, pp. 10–18, doi: 10.1145/1656274.1656278.

[30] Nivre, J.—Hall, J.—Nilsson, J.—Chanev, A.—Eryigit, G.—Kübler, S.—Marinov, S.—Marsi, E.: MaltParser: A Language-Independent System for Data-Driven Dependency Parsing. Natural Language Engineering, Vol. 13, 2007, No. 2, pp. 95–135, doi: 10.1017/s1351324906004505.

[31] Riaz, M.—Girju, R.: Another Look at Causality: Discovering Scenario-Specific Contingency Relationships with No Supervision. In 2010 IEEE Fourth International Conference on Semantic Computing (ICSC), 2010, pp. 361–368, doi: 10.1109/icsc.2010.19.

[32] Bach, E.: The Algebra of Events. In: Portner, P. H., Partee, B. H. (Eds.): Formal Semantics: The Essential Readings. Chapter 13. 2002, pp. 324–333, doi: 10.1002/9780470758335.ch13.

[33] Dölling, J.: Aspectual Coercion and Eventuality Structure. In: Robering, K. (Ed.): Events, Arguments, and Aspects: Topics in the Semantics of the Verbs. Studies in Language Companion Series, Vol. 152, 2014, pp. 189–226, doi: 10.1075/slcs.152.05dol.

[34] Moens, M.—Steedman, M.: Temporal Ontology and Temporal Reference. Computational Linguistics, Vol. 14, 1988, No. 2, pp. 15–28.

[35] Pustejovsky, J.—Castaño, J. M.—Ingria, R.—Saurí, R.—Gaizauskas, R. J.—Setzer, A.—Katz, G.—Radev, D. R.: TimeML: Robust Specification of Event and Temporal Expressions in Text. New Directions in Question Answering, Vol. 3, 2003, pp. 28–34.

[36] Pustejovsky, J.: The Syntax of Event Structure. Cognition, Vol. 41, 1991, No. 1-3, pp. 47–81, doi: 10.1016/0010-0277(91)90032-y.

[37] Saurí, R.—Knippen, R.—Verhagen, M.—Pustejovsky, J.: Evita: A Robust Event Recognizer for QA Systems. Proceedings of the Conference on Human Language Technology and Conference on Empirical Methods in Natural Language Processing, 2005, pp. 700–707, doi: 10.3115/1220575.1220663.

[38] Sprugnoli, R.—Tonelli, S.: One, No One and One Hundred Thousand Events: Defining and Processing Events in an Inter-Disciplinary Perspective. Natural Language Engineering, Vol. 23, 2016, No. 4, pp. 485–506, doi: 10.1017/s1351324916000292.

[39] De Swart, H.: Aspect Shift and Coercion. Natural Language and Linguistic Theory, Vol. 16, 1998, No. 2, pp. 347–385, doi: 10.1023/A:1005916004600.

[40] de Swart, H.—Verkuyl, H.: Tense and Aspect in Sentence and Discourse. ESSLLI Summer School, 1999.

[41] Vendler, Z.: Verbs and Times. Linguistics in Philosophy, Chapter 4. Cornell University Press, 1967, pp. 97–121.

[42] Sasse, H.-J.: Recent Activity in the Theory of Aspect: Accomplishments, Achievements, or Just Non-Progressive State? Linguistic Typology, Vol. 6, 2002, No. 2, pp. 199–271, doi: 10.1515/lity.2002.007.

Hasan Zafari received his B.Sc., M.Sc., and Ph.D. degrees in computer science in 2005, 2008, and 2017, respectively. Since 2008, he has been a faculty member in the Department of Computer Engineering at the Malayer Branch of Islamic Azad University. He has taught several courses on computer science over these years. His research interests include natural language processing, machine learning, deep learning, and data processing.

Maryam Hourali received her B.Sc. degree in mathematics as first rank from the University of Tehran in 2004, the M.Sc. degree in information technology from University of Science and Technology, Iran, in 2007, and the Ph.D. from Tarbiat Modares University in 2013. In 2012 she joined the Department of ICT, Malek-Ashtar University of Technology, Tehran, Iran. Her current research interests include AI, NLP, machine learning, text mining, and ontology.