# Coreference Resolution Using Verbs Knowledge

Hasan Zafari *
Department of Information and Communication Technology (ICT), Malek-Ashtar University of Technology, Tehran, Iran
hasan_zafari@yahoo.com
Maryam Hourali
Department of Information and Communication Technology (ICT), Malek-Ashtar University of Technology, Tehran, Iran
mhourali@mut.ac.ir
Heshaam Faili
School of Computer and Electrical Engineering, College of Engineering, University of Tehran, Tehran, Iran
hfaili@ut.ac.ir

Received: 11/Feb/2015 Revised: 07/Jan/2017 Accepted: 07/Mar/2017

**Abstract**

Coreference resolution is the problem of clustering mentions in a text that refer to the same entities, and is a crucial and difficult step in every natural language processing task. Despite the efforts that have been made to solve this problem during the past, its performance still does not meet today's application requirements. Given the importance of the verbs in sentences, in this work, we tried to incorporate three types of their information on coreference resolution problem, namely, selectional restriction of verbs on their arguments, semantic relation between verb pairs, and the truth that arguments of a verb cannot be coreferent of each other. As a needed resource for supporting our model, we generate a repository of semantic relations between verb pairs automatically using Distributional Memory (DM), a state-of-the-art framework for distributional semantics. This resource consists of pairs of verbs associated with their probable arguments, their role mapping, and significance scores based on our measures. Our proposed model for coreference resolution encodes verb's knowledge with Markov logic network rules on top of the deterministic Stanford coreference resolution system. Experiment results show that this semantic layer can improve the recall of the Stanford system while preserves its precision and improves it slightly.

**Keywords**: Coreference resolution, anaphora resolution, semantically related verbs, text inference, NLP

## 1. Introduction

Coreference resolution (CR) is determining mentions in the text which denote the same entity. By mention, we mean all pronouns, named entities and noun phrases that can refer to an entity. For example, there are many mentions in the following text snippet from which more important ones are marked with brackets.

> [Mexican football]$_{m1}$ got [a boost]$_{m2}$ in [September]$_{m3}$ when [former Brazil and Barcelona star]$_{m4}$ [Ronaldinho]$_{m5}$, joined [modest local club [Queretaro]$_{m6}$]$_{m7}$. [Carlos Trevino]$_{m8}$, [a former official of the [Queretaro state government]$_{m9}$]$_{m10}$, launched [an attack]$_{m11}$ on [Ronaldinho]$_{m12}$ before [the Brazilian]$_{m13}$ had played [a single game]$_{m14}$.

Two coreferent chains in the above text snippet are:

{m4, m5, m12, m13}, {m8, m10}

CR is an important subtask in natural language processing systems. Although it has become one of the core research topics in past decades, the complete solutions are elusive because it needs various types of knowledge to be solved completely.

This paper explores whether CR can benefit from verb's knowledge, especially semantic relation between verbs. More specifically, if we know that two verbs are semantically related, can we conclude that their arguments are semantically related too, and this relatedness leads us to conclude that they are corefer. The motivation comes from the fact that current CR systems are mostly relying on rather shallow features, such as the distance between the coreferent expressions, string matching, and linguistic form. It is expected that incorporating background knowledge in the form of semantically motivated features, along with an inference model incorporating it can improve the results.

In this paper, we focus on verb's knowledge and try to leverage it in CR. Verbs are one of the most important constituents in a text, which convey the main parts of sentence meaning and give a lot of information about it to the reader. Human can understand the meaning of sentence easily, using verb frame and prior knowledge about it. In other words, when a human reads a sentence, the first step in understanding the meaning is finding its verb, then using the information provided by the verb, and matching them with her previous experiences, the probable arguments and consequent events can be guessed.

Recently, researchers tried to use verb knowledge in CR and related subjects [1] [2] [3], [4] but it seems this line of research has more potentials and needs more attention.

In general, verbs provide three pieces of information that could be used in CR. The first piece is concerned with selectional restriction, which helps us to guess argument's semantic type. In other words, each verb sense has some predefined frames, and each frame slot

* Corresponding Author

could be filled by some limited entity types. For example, the verb overlook has two main senses, restricting subject and object to be a place (e.g. "The villa overlooks the town") or subject to be a person and object an abstract (e.g. "Nobody could overlook the fact"). The second piece has to do with the relationship between verbs in a text. That is, the verbs in a coherent text are related to each other semantically, creating a network of related arguments. These relationships could be used to detect coreferent mentions. For example, it may be useful to know that if someone joins a team, he may play for the team. This knowledge could be used to corefer [Ronaldinho]m5 to [the Brazilian]m13 in the above text snippet.

The third piece of information that we utilized in our model is the truth that arguments of a verb cannot be coreferent of each other. For example, in the sentence Carlos attacked he, we should have Carlos ≠ he.

The rest of the paper is organized as follows. We review related work in section 2, and propose our approach in section 3, including semantically related verbs acquisition, Predicate-Argument Structure and the model to applying this information to CR. Finally section 4 reports experiment results.

## 2. Related Work

Due to the importance of leveraging knowledge in natural language processing, many works have been developed to use them in CR.

Like us [5] extended the Stanford deterministic coreference system by linking mentions to Wikipedia. This process improved mention-detection and enabled new semantic attributes to be incorporated from Freebase. They tried to solve the named entity linking and coreference resolution, jointly.

In [6] YAGO is used to extract type relations for all mentions. These methods incorporated knowledge about all possible meanings of a mention. If a mention has multiple meanings, extraneous information might be associated with it.

Using named-entity linking in coreference resolution [7] extracted attributes from Wikipedia categories and used them as features in a mention-pair model. They reported 2 point improvements in B3 metric F1 points on non-transcript portion of the ACE 2004 dataset over their baseline system [8].

Caseframe network is proposed in [1] which is a kind of verbs knowledge for CR. A caseframe encodes two types of verb knowledge, including semantic category of verb's arguments and relationships between events. They extracted related verb's knowledge using easy to detect coreferent mentions. This approach requires a huge data to overcome the problem of extracting related verbs without sparsity. Our work, on the other hand, circumvents this problem by a novel idea (section 4).

In [3], [4] the authors tried to extract chains of events sharing a common participant. Again, because finding two verbs having coreferent arguments leads to likely sparse data of related verbs, their repository of related verb should be sparse relative to of our method. In order to estimate strength of pairwise relation, they used a distributional score based on how often two events share grammatical arguments, using pointwise mutual information (PMI). They predicted 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.

As a powerful inference method MLN has been employed as a common inference paradigm in many recent works. MLN can represent a probabilistic distribution over all possible configurations of the relations in an application, which is the case in NLP applications including CR. This key advantage of Markov logic causes its widespread use in CR.

In [9] the author encoded some basic knowledge about CR like appositive, mentions surface overlap, distance and so on in MLN rules and used Alchemy Toolkit for training and testing with Markov logic networks.

In [10] MLN is used in a pairwise coreference resolution model. They stated that their system gives a better performance than all the learning-based systems from the CoNLL-2011 shared task, and shows competitive performance compared with the best system from CoNLL-2011, which employs a rule-based method, on the same dataset.

Proposing a joint entity mention model for CR using MLN [11] used an anaphoricity classifier to discriminatively cluster mention with discourse entities. They reported a performance of 63.56% (gold mentions) on the official CoNLL 2012 data set.

A joint unsupervised approach using Markov logic is proposed in [12]. They offered a restricted set of entity level features. Clustering of mentions is driven by head features, and few semantic type and morphological features are used to assign further mentions to these clusters.

Recently [13] proposed a joint model, which despite other coreference models considers the mention head and boundary detection and coreference resolution jointly. Their main contribution is improving mention head and mention boundary detection, through which they improved end-to-end coreference decision. The only similarity between our method and this work is using head word of the mentions in both of the methods. We used head words of the mentions to determine their semantic category, while they proposed an ILP based method to determine head words more exactly that what is proposed by parsers and (without generalizing it) incorporating its knowledge in coreference decision. The main difference between our method and [13] is that our method is a knowledge based method which aims at resolving coreference decision that can be solved just using semantic knowledge of the verbs, while it tried to improve mention head and boundary detection that is useful in both coreference decision and final evaluation.

## 3. Proposed Approach

Our method aims at modelling CR using mention-pair approach, which uses knowledge of verbs in the form of a semantic layer on top of Stanford's deterministic coreference system [14]. This architecture is formulated as a joint inference problem using Markov Logic Networks [15]. It combines first-order logic formulas with probabilistic theory. This allows for transparent formalization of the method and flexibility of incorporating constraints and the feature set. Given the importance of the verbs in sentences, we tried to incorporate three types of their information on CR problem, namely, selectional restriction of verbs, semantic relation between verb pairs, and the truth that arguments of a verb cannot be coreferent of each other.
The main hypothesis of this paper is that the verbs which are semantically related to each other could have coreferent arguments. For example, it may be useful to know if someone has exploded, they may apologize at a later time.

***Ronaldinho** had only just arrived in town when **Trevino** was <u>exploded</u>. Following a wave of criticism, **he** <u>apologized</u> to the club and player.*

Given the fact that ***explode*** and ***apologize*** are semantically related, the pronoun *he* can be resolved to *Trevino* rather Ronaldinho which is not its correct antecedent. In order to enable the system to do such inference, some types of information about verbs and their relationships should be provided, including:

- In this context the verb "*explode"* has a patient (object) of type *person* and means "*show a violent emotional reaction*".
- Considering the meaning of context of "*explode"*, it is semantically related to "*apologize".*
- The object (patient) of "*explode"* is the subject (agent) of "*apologize"*.

This relationship can be of the different types such as causality, antonym, temporal and so on. Some example sentences containing coreferent mentions that could be understood using these relations are:

*CASUAL*: The police **shoot** <u>the theft</u> in the chest. <u>He</u> **died** immediately.

*TEMPORAL*: Firstly **defrost** <u>the meat</u> in the worm water. Then you can **cook** <u>it</u>.

*ANTONYM*: <u>The teen</u> **climbed up** the tree. <u>He</u> **climbed down** after retrieving his kite.

In the past, some efforts have been done to construct semantically related verbs repositories. For example, [16] tried to use some patterns to extract semantically related verbs from the web. Despite the value of these efforts, due to very low coverage and lack of required information such as semantic role mapping between verbs' arguments, the source could not be used in our method. WordNet [17] includes sematic relations such as cause, troponym and antonym for some verbs but this source also has both abovementioned problems (low coverage and lack of role mapping between verbs' arguments). The low coverage of WordNet relation is because of the fact that this resource does not include semantic relations those are plausible but not guaranteed. For example, it may be valuable to know that if someone has *shot* to a person, they may *kill* him. WordNet does not include the relation (shot $\rightarrow$ kill) since the event *shot* would not always cause *kill*. In [1] the author tried to find semantically related verbs by targeting verb pairs that have easy-to-find coreferent argument. It is a useful and accurate method, but could not lead to a comprehensive enough repository.

### 3.1 Semantically Related Verbs

Because of low coverage of available repositories of related verbs and lacking of needed information about related verbs such as probable common arguments and role mapping of common argument, they do not meet requirement of our method. As a result, we decided to create such a repository ourselves. For this purpose, we used Distributional Memory (DM) [18], a state-of-the-art framework for distributional semantics. We called this repository semantically related verbs repository (SRVR). It contains about 1M semantically related pairs of verbs associated with their probable arguments and significant scores denoting the relation strength, along with role mapping of their common argument.

The overall steps to creating this repository are as follows. Firstly, candidate tuples are extracted from DM. 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. That is, tuples where w1+link are phrasal verbs and tuples where w2 is non-action nominal. 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 DM, common arguments of semantically related verbs are extracted, which beside common argument weight (CAW), a measure of relations strength, can help to find mapping of verb pairs thematic roles. This process is depicted in figure 1 Part A.

#### 3.1.1 Extraction of Potential Relations

The DM tensor contains about 130M tuples automatically extracted from the 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 invers direction. We extracted all tuples of these 24 links from DM as initial tuples (*InitTuples*).

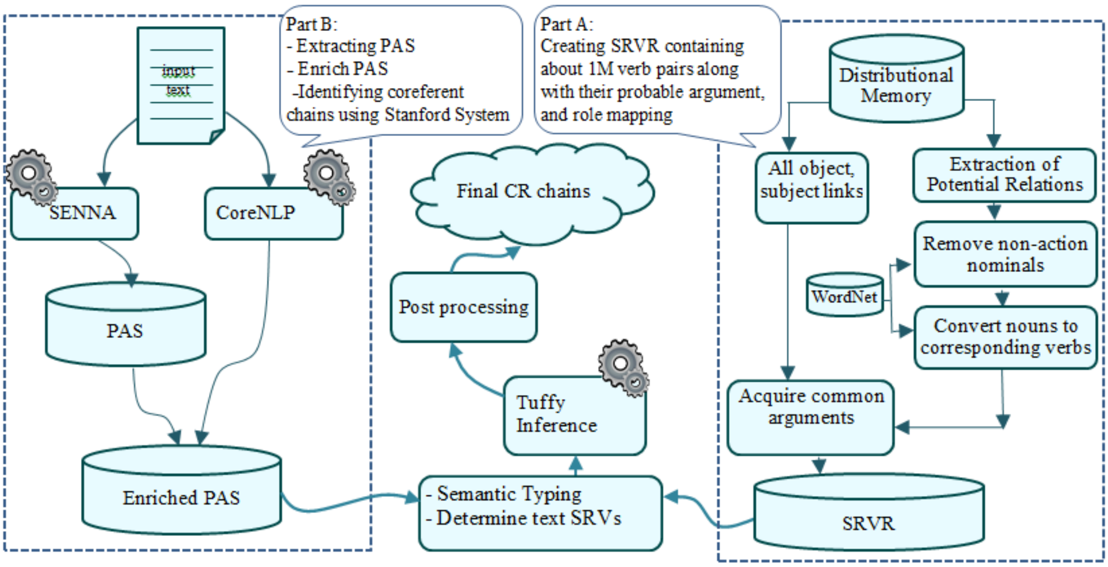


Fig 1. Proposed coreference resolution system

InitTuples totally include about 23M tuples in form of ⟨⟨w1, L, w2⟩, λ⟩. In these tuples w1 is mostly a verb (except coordination link) and w2 is always a noun. Table 1 shows these links along with their example tuples. For instance, in ⟨accuse, of, murder⟩, the preposition *of* is a sign of semantic relation between *accuse* and *murder*.

### 3.1.2 Removing Non -Action Nominals

Having extracted InitTuples from DM, the next step is to remove the tuples which do not contain event pairs from it. In a natural language, an event can be encoded using a verb or a noun. In all tuples ⟨⟨w1, link, w2⟩, λ⟩ extracted in previous subsection, w2 is a noun. Obviously, not all these nouns are event or action nominals. Following [19], action nominals are defined as "nouns derived from verbs (verbal nouns) with the general meaning of an action or a process". Also, according to [20], "an event is a situation that occurs or happens, 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 two types of tuples that do not contain event pairs, including (i) Tuples where w1 together with *preposition* create phrasal verbs like *account for*, and (ii) Tuples where based on WordNet event denoting synsets w2 is not event at all, like *day*

In order to remove phrasal verbs from the InitTuples, we used a predefined list of phrasal verbs to remove such tuples from InitTuples.

For detecting and removing tuples where w2 is not an event at all, we used the WordNet hypernymy structure. For this purpose we have chosen five WordNet synsets including {event, process, state, message, symptom} so that their hyponym (children) are mostly action nominals. For example, "event" is such a synset and has many action nominals as its hyponym (children) e.g. *attack*, *discovery*, *strike*, *escape*. Beside *event*, there are other synsets that can denote an event like *process*. We have chosen these five synsets from WordNet by examining about 500 event nouns. 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.

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

| 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 | ⟨discount, coord-1, price⟩ |

#### 3.1.3 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 verbs, 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) =
{crop, plant, tree, grape, seed, potato, grain, fruit, wheat }

Argument (plant) $\cap$ Argument (crash) = $\varnothing$

We call these words that can be arguments of both verbs as common arguments. Common arguments can be found in *subject* and *object* links in DM. There are more than 10M such links in DM. We define Common Argument Weight (CAW) as the relative measure of the strength between two verbs. Algorithm 1 has three outputs, including CAW, common arguments and role mapping. Common arguments can be acquired by selecting common arguments (CA) of top n tuples from *sortedTuples*, sorted tuples of `jointTuples` having the highest value of f (λ1, λ2). 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 of the 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 one 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 *subject* of *escape* and *object* of *arrest* most of the times. So, the mapping *escape* (sbj) = arrest (obj) is obtained for this verb pair.

Although gathered from large parsed corpora and not necessarily co-occurring in the same document, the acquired common arguments are so accurate. Beside a metric for relations strength measurement, common arguments can act as a means to disambiguate polysemous verb with respect to another verb.

**Input**: verb1, verb2
**Outputs**: sbj, obj mapping, CAW, common arguments
`/*tuples schema:⟨verb,link,arg,λ⟩ */`
V1Links ← sbj, obj links of verb1 in DM
V2Links ← sbj, obj links of verb2 in DM
jointTuples ← empty
**for** t1∈ V1Links **do**
**for** t2∈ V2Links **do**
**if** t1.arg = t2.arg **then**
CA ← t1.arg1
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 ← $\pi_{\lambda c}(\sigma_{top1}$ sortedTuples)
RoleMapping ← $\pi_{rel1, rel2}(\sigma_{top1}$ sortedTuples)
commonArgs ← $\pi_{CA}(\sigma_{top\ n}$ sortedTuples)

Algorithm 1. Extracting CAW, Common Arguments and Role Mapping.

For instance, in pair (install, execute), common arguments are words denoting a program or script, which indicates execute means run a program, but in pair (arrest, execute) common arguments denote a prisoner or criminal, which indicates execute means put to death. Table 2 shows some examples of CAW, common arguments and role mapping.

To acquire this information for two given verbs verb1 and verb2, we have firstly chosen *subject* and *object* links (tuples) of DM for them, namely `V1Links` and `V2Links`, respectively. There are about a few thousands of such links for each verb in DM. 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 × V2Links` if `V1Links.arg = V2Links.arg`, we keep (join) them, otherwise discard them. Then, we calculate f (λ1, λ2) as a function of λ1 and λ2 of the joint tuple, where λ1 is the weight of verb1 tuple and λ2 is the weight of verb2 tuple.

Table 2. some examples of SRVR record data

| Verb pair | Common arguments | Role mapping | CAW |
|---|---|---|---|
| design- print | page, poster, card, form, leaflet, book, work, logo, character, map, stamp | obj-obj | 685.7961 |
| sow- harvest | crop, seed, field, grain, plant, corn, wheat, bean, barley, onion | obj-obj | 651.5708 |
| rob-arrest | man, gang, thief, youth, pirate, bandit, criminal, robber, soldier, guy, burglar | 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 |

Lastly, by sorting the joint tuples based on f ($\lambda 1$, $\lambda 2$) in descending order and picking the highest one, `CAW` can be calculated as a function of $\lambda 1$, $\lambda 1$ of this tuple. See algorithm 1 for more details.

## 3.2 Predicate Argument Structure (PAS)

PAS is a concise notation for representing a text document, since it constitutes events within the text and their participants. Understanding events and their participants is crucial in order to semantically analyze the natural language text. The event is usually described by a verb. The participants in this event are noun phrases or pronouns, each of which has a specific role in the event. Semantic roles are representations that express the abstract role that arguments of a predicate can take in the event. These roles are diverse but in this article, we are only interested in agent and patient who are usually equivalent to the syntactic role of subject and object, respectively.

The relationship between the verb and its arguments is important to us for two reasons. First, by knowing the verb and the syntactic (or semantic) relation of its argument, we can usually determine the semantic categories of the argument. On the other hand, to properly map the common argument of two related verbs, we need to know what relation (agent or patient) each argument has with its corresponding verb. It helps us identify the coreferent argument of two related verbs. As a result, it is necessary to process the input text and extract all of its predicate-argument structures. We used an open-source SRL tool SENNA [21] to convert input text to PAS.

### 3.2.1 Enriching PAS

Extracting PAS from text accounts for mention extraction of the text because mentions are actually arguments in the PAS. The predicates and the semantic relationship of arguments with them are some features of our model. Nevertheless, in our CR model, we need another information of the mentions that are not provided in PAS now, such as the exact boundaries of the mentions, head noun of the noun-phrase mentions, semantic category of the mentions, and coreferent mentions that can be detected using a state of the art CR system. We intended to enrich PAS with this information.

We used Stanford CoreNLP [22] to extract our needed information from the input text. Stanford CoreNLP is an NLP package which processes the input text. It provides a set of natural language analysis, including noun phrase extractor, head finder, named entity extractor and deterministic Coreference resolution. We used Stanford deterministic CR tool [14] to determine currently known coreferent mention of the text. Actually, in this article we are trying to add a semantic layer to this layered CR system. Then, in PASs, every mention is replaced with its representative (if any). Stanford deterministic CR tool was the top-ranked system at the CoNLL-2011 shared task. However, this system cannot find all coreferent mentions, and not all of its output is correct. Yet, it is useful in our method because it could resolve easy-to-find coreferent mentions almost correctly, which is very valuable for us. At the end of this step, all named entities of the input text are extracted to be consulted with output of SENNA named entity extractions, as explained in the next section.

### 3.2.2 Semantic Typing

Semantic typing is to accurately determine the conceptual categories of mentions, which we call Semantic Category (SC). SC has an important role in our method. First, it is one of the mentions features in CR model that prevents linking non-coreferent mentions. If the SC of the mention is known, most of the non-coreferent antecedents can be ruled out. Secondly, it helps us select just those related verbs from SRVR that are consistent with the given verb in the text. In other words, two verbs may be related with respect to some of their meanings and not related considering their other meanings. One such example is pair (charge – disconnect) that is semantically related if their common argument is *a devise or battery* (i.e. charge means to *energize a battery*) and (charge- arrest) that are semantically related if their common argument is a *person* (i.e. charge means *blame for*).

In an ideal case, SC would be able to discriminate between different types of mentions, while not considering similar (coreferent) mentions different (overfitting). SC is a set of predefined categories that determines the general type of a mention. It is like categories in NER with the difference that it tries to solve its shortcomings. One shortcoming of NER is its over-simplified ontological model, leaving instances of other potentially informative categories unidentified. Hence, the utility of named entity information is limited. In addition, instances to be detected are mainly restricted to proper nouns, while we are mainly facing with common nouns in text.

SC can be fine or coarse-grained. In general, coarse-grain SC can be like categories that are used in NER, such as the persons, organizations, locations, etc. fine-grain SC can be the head word of the mention (noun phrase) or its hypernym (parent) in the WordNet (any synset in the WordNet). In this study, we are looking for SCs that are neither too coarse nor too fine-grained. That is, the set of categories used in semantic typing must be adequate enough to serve the tasks. Too coarse-grained SC may not be able to distinguish different mentions very well. Too fined-grained SC, on the other hand, could introduce different categories for coreferent mentions.

Table 3: selected semantic categories from WordNet

| SC | Nouns denoting | SC | Nouns denoting |
|---|---|---|---|
| Animal | animals | object | natural objects (not man-made) |
| Artifact | man-made objects | quantity | quantities and units of measure |
| Attribute | attributes of people and objects | phenomenon | natural phenomena |
| Body | body parts | plant | plants |
| Cognition | cognitive processes and contents | possession | possession and transfer of possession |
| communication | communicative processes | process | natural processes |
| Event | natural events | person | people |
| Feeling | feelings and emotions | relation | relations between people or things |
| Food | foods and drinks | shape | two and three dimensional shapes |
| Group | groupings of people or objects | state | stable states of affairs |
| Location | spatial position | substance | substances |
| Motive | goals | time | time and temporal relations |

In this paper, we adapted SC introduced in [23] which extended the named-entity recognition approach to the classification of common nouns into 26 different supersenses. Rather defining these categories manually, they adopted the "lexicographer class" labels used in WordNet, which include labels such as person, location, event, quantity, etc. Table 3 shows these categories.

### 3.2.3 Assigning SC to the Mentions

There are various sources to assign SC to a mention but usually none of these sources are able to determine SC solely. That's why we tried to combine them to determine SC. These resources include:

- **Mentions head word**: the head word of a noun phrase can be obtained using Stanford's dependency parser. The head words itself can be considered as a fine-grain SC. However, by generalizing it using WordNet hypernym structure, it can be converted into coarser grained SC. In *former Barcelona player*,

for instance, the head word is *player*, which yields *person* after generalizing using WordNet hypernym structure with respect to table 3 categories.

- **Pronoun**: For many pronouns, SC is known. For personal pronouns (he, she, her, you, and so on) for example, the SC is *person* and for the locative pronoun (here, there, somewhere) is *place*, and for temporal (now, then, sometimes) is *time*.
- **Coreference chains** acquired from Stanford deterministic system: Since all of the mentions in a coreference chain refer to same entity, they should have the same SC. Therefore, given a coreference chain, if the SC of one of its mention is known, we can assign it to all other mentions in that chain. For example, in a chain like {x, he}, knowing the SC of the pronoun *he* is *person*, the SC of mention x will be *person* too.
- **Selectional restriction**: The predicate selectional restriction on its argument is a good source to determine SC of the arguments. Given the predicate and semantic (or syntactic) relation of its argument, we can use DM to determine the arguments SC. For example, in *eat (obj: x)* the SC of x is likely *food*.
- **NER**: Named Entity Recognition tools map named entity to one of the predefined categories like person, location, etc. we used NER of the SENNA and Stanford CoreNLP as one of the sources in determining mentions SC.

Due to the different accuracy of any of the resources listed above, the order of applying them should be so that the more precise ones are examined before the less precise resources. This order is as follows:

1. For an unambiguous pronoun the SC is assigned according to its category.
2. For mentions x that their head word has just one sense, the SC (x) assigned using hypernym structure of WordNet (algorithm 2). Otherwise, CatsHead (x), list of all probable SCs of x along with their weights using algorithm 2 is created.
3. If the mention x is a member of a coreference chain $\{m_1, \ldots, m_n\}$, and the SC of one of the $m_i$ is y then the SC of x will be y.
4. For the named entities that NER of SENNA and Stanford CoreNLP are unanimous for, the SC equals to the NER category. Why not being limited to only one of these sources is because these tools usually have errors (especially between *person* and *organization*).
5. For the mention x that selectional restriction introduces just one SC, calculate and assign SC. Otherwise, create CatsPred (x), list of all probable SCs of x with respect to selectional restriction of its predicate using DM.

For the mentions that SC is not determined using the above steps, combine SCs introduced by CatsHead and CatsPred. CatsHead and CatsPred include all probable SCs for a mention along with the weight of each SC. The weight of each SC in CatsHead is calculated and accumulated based on the order of senses in WordNet. That is, for each sense with sense-number i a coefficient, z (i), is used to calculate its weight. For CatsPred, the weight is calculated using the tuple weight, $\lambda$. Then by merging these two lists and choosing the most likely SC, the mention's SC will be determined.

In the merging process, the lists join together with respect to their SC, and the weights are multiplied. Then the multiplied weights are sorted and the SC that has the highest value is selected as the mentions SC.

```
Input: Noun
Outputs: CatsHead
1: SCs ← all SCs in the table 3
2: for each sc in SCs do
3:    w (sc) ← 0
4:  maxSense ← number of noun Senses (Noun)
5:  for s# =1 to maxSense do
6:     for each sc in SC do
7:        if sc ∈ hypernym (s#) then
8:           w (sc) +=z (s#)
9:  for each sc in SCs do
10:      if w (sc) >0 then
11:        append (CatHead, sc, ″/″, w (sc))
12: return CatHead
```

Algorithm 2. Extracting mention category based on WordNet hypernym structure.

For example, suppose we want to calculate SC for the mention *his first match* in the sentence, *he played his first match*. The head word of this mention is *match*. Its SC is not calculable by none of the 1 to 5 items listed above. Therefore, we have to calculate its SC by merging its CatsHead and CatsPred. Figure 2 shows the value of these lists.

**Input text:** He played his first match
**Target mention**: his first match
**PAS**: play (A1: his first match)
**Head word**: match
**CatHead**:
artifact/13-event/5-person/4-amount/2-cognition/2 -group/1
**CatsPred**:
event/7.8-artifact/1.7 -attribute/8.5 - animal/6.8- state/2.7
**Results**: SC (match) = *event*

Fig 2. Calculating SC by merging CatsHead and CatsPred

### 3.2.4 Related Verbs and Meaning in Context

SC has another usage in our method, i.e. identifying the correct sense of two verbs in the input text that deemed related with respect to the SRVR. In fact, a verb in the input text could have supposed to be related to many other verbs in that text. On the other hand, these relations may hold with respect to some senses of those verbs but not hold with respect to other senses of them. For instance, the pair *arrest-execute* is semantically related if *execute* means *put to death*, but not related to each other if *execute* means *run a program*. We used SC of the common arguments of two related verbs in the SRVR to determine if they are really related in the input text or not. Figure 3 shows a text contains two verbs, *execute* and *test*.

This pair is deemed semantically related according to SRVR, but is not related based on their meaning in the context. The common arguments of this verb pair according to SRVR are program, code, procedure, process, and function with SC of communication. On the other hand, the argument of execute in the text is prisoner with SC of person. Since person and communication are not compatible, we conclude that execute and test are not related here.

**Input text:**
The prisoner was executed …. We have tested all the doors' locks.
**KB**: execute→test
**Common arguments**:
program, code, procedure, process, function
SC(Common arguments) = *communication*
**Results**:
Execute (*person*) ≠ execute (*communication*)

Fig 3. Using Common arguments of the related verbs to prevent incorporating a pair of related verbs from SRVR that is not related considering their meaning in the context

## 3.3 Applying Verb Knowledge to CR

In this section, we propose our unsupervised model for CR. The model is a knowledge-based model which incorporates verb knowledge into a CR system. We incorporated three types of verb's information on CR problem. The model is based on Markov Logic Networks [24] which combines first-order logic rules with probabilistic theory. This allows encoding different types of features and constraint in CR easily and in a way that can be understood by people.

### 3.3.1 Markov Logic Network

Markov Logic Network (MLN) is a probabilistic extension of first-order logic, which becomes one of the most powerful tools for joint inference. MLN specifies what data (evidence) is available, what predictions to make (query), and what constraints and correlations exist (rules). The process of computing predictions given an MLN is called inference. In MLN, one can write first-order logic rules with weights. The weight of a rule specifies its confidence. This allows one to capture rules that are likely, but not certain, to be correct. This is the case for most of the constrains and rules that hold between mentions' features and their coreference status

### 3.3.2 Coreference Model

In this section, we explain details of the MLN based model for CR. Specifically, we express rules of our method in terms of query and evidence predicates.

The evident predicates which provide information for the rules include extracted predicate from the text, related verbs from SRVR, and semantic type of mentions. These predicates are expressed in the following schema.

```
1:  vA0(verb,mention)
2:  vA1(verb, mention)
3: *relatedA0A0(verb,verb,float wgt)
4: *relatedA0A1(verb,verb,float wgt)
5: *relatedA1A0(verb,verb,float wgt)
6: *relatedA1A1(verb,verb,float wgt)
7: *vA0A1(verb,mention,mention)
8: *Type(mention,type)
9:  coref(mention,mention)
```

The main query predicate is `coref(x,y)` which is true if x is corefer with y.

MLN rules specify the relationship between evidence features and the query predicate.

The following rule reflects the fact that if according to the evidence, two predicates of the input text are related, and the mapping between their common arguments is A0A0 (i.e. the agent of the first verbs is mapped to the agent of the second verb), then the agent argument of them could be coreferent to each other.

```
10: wgt:vA0(v1,m1),vA0(v2,m2),
relatedA0A0(v1,v2,wgt)=> coref(m1,m2)
```

There are three rules corresponding to the other three possible mappings,:

```
11: wgt:vA0(v1,m1),vA1(v2,m2),
relatedA0A1(v1,v2,wgt)=> coref(m1,m2)
12: wgt:vA1(v1,m1),vA0(v2,m2),
relatedA1A0(v1,v2,wgt)=> coref(m1,m2)
13: wgt:vA1(v1,m1),vA1(v2,m2),
relatedA1A1(v1,v2,wgt)=> coref(m1,m2)
```

The weight of these rules is equal to the value of relation strength between the verb pair in SRVR.

Another knowledge that verbs provide to us is that if two mentions have the same arguments, they should not be coreferent. This is ensured by the hard rule (which has infinite weight and must be satisfied)

```
14: vA0A1(v1,m1,m2)=>!coref(m1,m2).
```

Mentions with different SC cannot be coreferent. This is ensured by the hard rule (which has infinite weight and must be satisfied)

```
15: Type(m1,t1),Type(m2,t2),[t1<>t2]
  =>!coref(m1,m2).
```

General rules of reflexivity, symmetry, and transitivity of the model are:

```
16: coref(x,x).
17: coref(x,y)=> coref(y,x).
18: coref(x,y),coref(y,z)=>coref(x,z).
```

### 3.4 Implementation and Post Processing

We used a state-of-the-art MLN systems, namely Tuffy [25] in order to implement MLN rules. It is implemented in Java and used PostgreSQL as the underlying database system. Tuffy processes the evidence, program, and query files, and produces output links along with their probabilities. In order to make Tuffy to include probabilities of the output, we set the marginal inference. Then we have to merge these links to get final coreference chains.

It should be noted that the evidence that provides information for inference is not free of errors. Rather, there are errors in almost all the past steps that have been done to provide evidence. Hence, the inference results may contain mistaken output links. The post processing step aims at concatenating the output links to get coreference chains while tries to reduce (eliminate) mistaken links. For this purpose, a greedy algorithm is employed, which looks like the maximum spanning forest algorithm.

The algorithm receives weighted output links of the inference as input and produces coreference chains. To do so, a graph G (V, E) is created; where V represents all the mentions in the input text, and E determines the mention pairs that are coreferent. E is initially empty (the graph has no edge at first). The following loop examines inference output (L) one-by-one and adds them to the set E of the graph G (V, E) if they do not have conflict with the existing rules.

At the termination of the algorithm, the forest forms coreference chains of the document mentions.

*While L is not empty*

1. *Remove an edge with maximum weight from L*
2. *If the removed edge connects two trees which do not contains mention(s) that conflict with rules 14 and 15,*
    a. *Add it to the graph G.*
    b. *Combine two trees into a single tree.*

## 4. Experiments

In this section, the results of the proposed coreference system will be evaluated with respect to the baseline, i.e. Stanford's deterministic coreference system. For this purpose, two separate test data are used. The first is CoNLL 2012 coreference data. The second test set is about odd news of Yahoo News (www.yahoo.com/news/odd/). The reason for selecting odd Yahoo News is that this news is mainly about the strange happenings which contain many events. Hence the proposed method can be better evaluated on this data.

### 4.1 Experimental Setup

**Datasets**: since the aim of our approach is to resolve CR on the text that needs semantic knowledge of related verbs, we have to evaluate our method on the data that need such knowledge. As the standard test data for CR, like CoNLL 2012 data, may not include such text, we collected news wire documents that talk about news containing events. Odd news of the Yahoo site has such a property. Hence, we collected 20 news documents from this site to create second test data. Two human annotators were asked to manually annotate this data set. The human inter-annotator agreement achieved on this test set is 93%.

**Baseline System**: We choose two publicly available state-of-the-art end-to-end coreference systems as our baselines: *Stanford* coreference resolution system [14], winner of the shared task 2011, and *Illinois* coreference system [13]. The first which our method is implemented on top of it, is a rule based system implemented as part of Stanford CoreNLP toolkit [22]. It comprises a pipeline of "sieves" that merge coreferent mentions according to deterministic rules. Higher precision sieves are applied earlier in the pipeline according to the following order, looking at different aspects of the text, including: (1) speaker identification, (2-3) exact and relaxed string matches between mentions, (4) precise constructs, including appositives, acronyms and demonyms, (5-9) different notions of strict and relaxed head matches between mentions, and finally (10) a number of syntactic and distance cues for pronoun resolution.

Table 4: Performance of coreference resolution for all systems on the test set 1

| | MUC | | | $B^3$ | | | BLANC | | |
|---|---|---|---|---|---|---|---|---|---|
| System | R | P | F1 | R | P | F1 | R | P | F1 |
| Stanford | 59.66 | 61.54 | 60.6 | 55.09 | 49.2 | 51.97 | 55.14 | 48.11 | 51.34 |
| Illinois | 52.33 | 57.25 | 54.68 | 45.33 | 60.52 | 51.83 | 52.00 | 68.42 | 59.09 |
| Our system | 61.25 | 62.08 | 61.66 | 55.73 | 49.03 | 52.16 | 56.29 | 48.08 | 51.86 |

Table 5: Performance of coreference resolution for all systems on the test set 2

| | MUC | | | $B^3$ | | | BLANC | | |
|---|---|---|---|---|---|---|---|---|---|
| System | R | P | F1 | R | P | F1 | R | P | F1 |
| Stanford | 50.0 | 83.33 | 62.5 | 38.09 | 90.0 | 53.53 | 25.0 | 88.88 | 39.02 |
| Illinois | 42.06 | 65.51 | 51.23 | 34.89 | 69.78 | 46.52 | 27.08 | 81.21 | 40.62 |
| Our system | 65.41 | 81.25 | 72.47 | 45.81 | 82.06 | 58.80 | 31.54 | 90.9 | 46.83 |

**Evaluation Metrics**: we used widely recognized metrics MUC, B3, and BLANC.

**MUC** [26] Link-based metric which measures how many predicted and gold clusters need to be merged to cover the gold and predicted clusters, respectively.

**B3** [27] Mention-based metric which measures the proportion of overlap between predicted and gold clusters for a given mention.

**BLANC** [28] Metric based on the Rand index [29] that considers both coreference and non-coreference links to address the imbalance between singleton and coreferent mentions.

It should be noted that we used system mention to evaluate our method; that is, the detection of the mention boundaries has been done using the system.

**Results:**

Table 4 and 5 compare the performance of our system against the baseline systems with respect to Test-set 1 and 2, respectively. As expected, the performance of the proposed system on the Test-set 2 which contains more events is more evident. The reason is that the baseline systems just uses shallow features, which are not sufficient to identify coreferent mentions in such documents. Practically, coreferent mentions in such documents has a different surface, implying different aspects of referring entity. For person entity, for example, referring expressions may refer to its name, nationality, job, role within the story (victim, criminal, passenger, etc.), and so on. Hence resolving such a different surface mentions need more semantically motivated features like what we utilized.

## References

[1] D. Bean and E. Riloff, "Unsupervised Learning of Contextual Role Knowledge for Coreference Resolution," Proc. Hum. Lang. Technol. Conf. North Am. Chapter Assoc. Comput. Linguist. (HLT-NAACL 2004), pp. 297–304, 2004.

[2] H. Lee, M. Recasens, A. Chang, and M. Surdeanu, "Joint entity and event coreference resolution across documents," in Association for Computational Linguistics, 2012.

[3] N. Chambers and D. Jurafsky, "Unsupervised Learning of Narrative Schemas and their Participants," Proc. Jt. Conf. 47th Annu. Meet. ACL-IJCNLP 4th Int. Jt. Conf. Nat. Lang. Process. AFNLP, vol. 2, no. August, p. 602, 2009.

[4] N. Chambers and D. Jurafsky, "Unsupervised learning of narrative event chains," Proc. Assoc. Comput. Linguist., vol. 31, no. 14, pp. 789–797, 2008.

[5] L. Zilles and D. S. Weld, "Joint Coreference Resolution and Named-Entity Linking with Multi-pass Sieves," Emnlp, no. October, pp. 289–299, 2013.

[6] A. Rahman and V. Ng, "Coreference Resolution with World Knowledge," Acl, no. June, pp. 814–824, 2011.

[7] L. Ratinov and D. Roth, "Learning-based multi-sieve co-reference resolution with knowledge," Proc. 2012 Jt. Conf. Empir. Methods Nat. Lang. Process. Comput. Nat. Lang. Learn., no. 1, pp. 1234–1244, 2012.

[8] E. Bengtson and D. Roth, “Understanding the value of features for coreference resolution,” Proc. Conf. Empir. Methods Nat. Lang. Process. - EMNLP ’08, vol. 51, no. October, p. 294, 2008.
[9] S. Huang, Y. Zhang, J. Zhou, and J. Chen, “Coreference Resolution using Markov Logic Network,” Science (80-. )., pp. 157–168, 2009.
[10]Y. Song, J. Jiang, W. X. Zhao, S. Li, H. Wang, and 王厚峰, “Joint learning for coreference resolution with Markov logic,” Proc. 2012 Jt. Conf. Empir. Methods Nat. Lang. Process. Comput. Nat. Lang. Learn., no. July, pp. 1245–1254, 2012.
[11] T. Bögel and A. Frank, “A joint inference architecture for global coreference clustering with anaphoricity,” Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 8105 LNAI, pp. 35–46, 2013.
[12] H. Poon and P. Domingos, “Joint Unsupervised Coreference Resolution with Markov logic,” Proc. Conf. Empir. Methods Nat. Lang. Process., no. October, p. 650, 2008.
[13] H. Peng, K.-W. Chang, and D. Roth, “A Joint Framework for Coreference Resolution and Mention Head Detection,” CoNLL, pp. 12–21, 2015.
[14] H. Lee, Y. Peirsman, A. Chang, N. Chambers, M. Surdeanu, and D. Jurafsky, “Stanford ’ s Multi-Pass Sieve Coreference Resolution System at the CoNLL-2011 Shared Task,” Proc. Fifteenth Conf. Comput. Nat. Lang. Learn. Shar. Task. Assoc. Comput. Linguist., pp. 28–34, 2011.
[15]M. Richardson and P. Domingos, “Markov logic networks,” Mach. Learn., vol. 62, no. 1–2 SPEC. ISS., pp. 107–136, 2006.
[16] T. Chklovski and P. Pantel, “VerbOcean: Mining the Web for Fine-Grained Semantic Verb Relations.,” EMNLP, no. Lin 1997, 2004.
[17] G. A. Miller, “WordNet: a lexical database for English,” Commun. ACM 38.11 39-41., 1995.
[18] M. Baroni and A. Lenci, “Distributional Memory: A General Framework for Corpus-Based Semantics,” Comput. Linguist., vol. 36, no. 4, pp. 673–721, 2010.
[19] B. Comrie, “The syntax of action nominals: A cross-language study,” Lingua, vol. 40, no. 2–3, pp. 177–201, 1976.
[20] M. Pustejovsky, J., Hanks, P., Sauri, R., See, A., Gaizauskas, R., Setzer, A., Radev, D., Sundheim, B., Day, D., Ferro, L. and Lazo, “The timebank corpus,” Corpus Linguist., vol. 40, pp. 647–656, 2003.
[21] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, “Natural Language Processing (almost) from Scratch,” J. Mach. Learn. Res., vol. 12, pp. 2493–2537, 2011.
[22] C. D. Manning, J. Bauer, J. Finkel, S. J. Bethard, M. Surdeanu, and D. McClosky, “The Stanford CoreNLP Natural Language Processing Toolkit,” Proc. 52nd Annu. Meet. Assoc. Comput. Linguist. Syst. Demonstr., pp. 55–60, 2014.
[23] M. Ciaramita and Y. Altun, “Broad-coverage sense disambiguation and information extraction with a supersense sequence tagger,” Proc. 2006 Conf. Empir. Methods Nat. Lang. Process. - EMNLP ’06, no. July, pp. 594–602, 2006.
[24]M. Richardson and P. Domingos, “Markov logic networks,” Mach. Learn., vol. 62, no. 1–2, pp. 107–136, 2006.
[25] F. Niu, C. Ré, A. Doan, and J. Shavlik, “Tuffy: Scaling up Statistical Inference in Markov Logic Networks using an RDBMS,” Proc. VLDB Endow., vol. 4, no. 6, pp. 373–384, 2011.
[26] M. Vilain, J. Burger, J. Aberdeen, D. Connolly, and L. Hirschman, “A Model-Theoretic Coreference Scoring Scheme,” Messag. Underst. Conf., no. 1, pp. 45–52, 1995.
[27] A. Bagga and B. Baldwin, “Algorithms for scoring coreference chains,” in The First International Con-ference on Language Resources and Evaluation Work-shop on Linguistics Coreference, 1998, pp. 563–566.
[28] Marta Recasens and Eduard Hovy, “BLANC: Implementing the Rand index for coreference evaluation,” Nat. Lang. Eng., vol. 17(4), pp. 485–510, 2011.
[29] W. M. Rand, “Objective criteria for the evaluation of clustering methods,” J. Am. Stat. Assoc., vol. 66(336), pp. 846–850, 1971.

**Hasan Zafari** received his B.Sc. degree in Software Engineering from University of Kerman, Iran. He received his M.Sc. degree in Software Engineering from Islamic Azad university of Arak, Iran. He is currently a Ph.D. student at the Department of Information and Communication Technology (ICT), Malek-Ashtar University of Technology, Tehran, Iran.

**Maryam Hourali** is an Assistant Professor at Malek-Ashtar University of Technology, Iran. She received her B.Sc. in Applied Mathematics in university of Tehran, Iran, and her M.Sc. in Information Technology from the Iran University of Science and Technology (IUST), and her Ph.D. in Industrial Engineering-Information Technology from Tarbiat -Modares University, Iran.

**Heshaam Faili** is an Associate Professor at School of Electrical and Computer Engineering, university of Tehran, Iran. He received his B.Sc. and M.Sc. in Software Engineering, from Sharif University of Technology and his Ph.D. in Artificial Intelligence, from Sharif University of Technology.