Eliciting Implicit Evocations Using Word Embeddings and Knowledge Representation

Sébastien Harispe, Medjkoune Massissilia, Jacky Montmain

Eliciting Implicit Evocations using Word Embeddings and Knowledge Representation

Sébastien Harispe[1], Massissilia Medjkoune[1], and Jacky Montmain[1]

École des mines d'Alès, LGI2P Research Center,
Parc scientifique G. Besse 30035 Nîmes Cedex 1, France
firstname.lastname@mines-ales.fr

Abstract. Automatic elicitation of implicit evocations - i.e. indirect references to entities (e.g. objects, persons, locations) - is central for the development of intelligent agents able of understanding the meaning of written or spoken natural language. This paper focuses on the definition and evaluation of models that can be used to summarize a set of words into a unique unambiguous entity identifier selected from a given ontology; the ability to accurately perform this task being a prerequisite for the detection and elicitation of implicit evocations on spoken and written contents. Among the several strategies explored in this contribution, we propose to compare hybrid approaches taking advantages of knowledge bases (symbolic representations) and word embeddings defined from large text corpora analysis. The results we obtain highlight the relative benefits of mixing symbolic representations with classic word embeddings for this task.

1 Introduction

Developing automatic approaches enabling human spoken and written productions to be deeply understood is central for the development of artificial agents capable of complex human-machine interactions and collaborations. This broad challenge, largely studied by the Artificial Intelligence community[1], aims at developing approaches capable of *capturing* the meaning conveyed by units of language (from word utterances to sequences of phrases); this is central for numerous processes widely studied in the literature: Question Answering, Information Extraction and Information Retrieval, among others.

In this paper, we focus on studying aspects tightly related to the development of approaches for understanding the meaning and semantics of large units of languages such as sentences or paragraphs. The positioning of our work is, broadly speaking, closely related to Named-Entity Recognition (NER), i.e. detection of explicit entity mentions in texts [12]. We are more particularly interested in fine-grained entity recognition, not only aiming at detecting classes of entities as it is classically done in NER by detecting references to persons or locations for instance. We are here rather interested in entity linking, i.e. at linking specific

[1] e.g. in the Natural Language Processing and Computational Linguistics domains.

unambiguous identifiers provided by knowledge bases - such as DBpedia and Yago Uniform Resource Identifiers (URIs) or WordNet synset identifiers [1,11] - to entities mentioned into texts [10]. We study in particular the problem of eliciting implicit evocations, i.e. references to unambiguous entities that are not mentioned by lexical forms of those entities. As an example, in the sentence "*I've visited the capital of Spain as well as Picasso birthplace city last summer*", the utterances '*capital of Spain*' refers to `Madrid`, '*Picasso birthplace city*' to `Málaga`. Similarly, telling you that "*this morning I've eaten a yellow tropical fruit very much liked by monkeys*" should give you a good idea of the kind of fruit I've eaten (i.e. a `Banana`).

The aim of this paper is to study automatic approaches that are able to elicit implicit evocations; similarly to the way humans are most often able to understand them. Developing approaches enabling such a process is important for capturing the meaning of units of languages; their direct applications for semantic indexing and information retrieval, as well as their indirect potential applications to question answering, information extraction, topic identification or sentiment analysis to cite a few, are numerous. Due to the breadth and complexity of the task, we here focus on eliciting implicit evocations considering the words mentioning the entity evocation to be given, e.g. considering a bag of words extracted from the initial sentence {'*yellow', 'tropical fruit', 'monkeys'*} we expect the approach to identify the entity `Banana`.[2] We also consider that there is no need to take into consideration contextual information for detecting the implicit evocation - otherwise stated, the knowledge base that is required to answer the question is therefore considered to be static and not contextual.

The paper is organized as follows; Section 2 presents related works as well as the fundamental notions on which will be based our contributions. Section 3 introduces the different models that can be used to detect implicit entity evocations from bags of words. Section 4 presents the evaluation protocol as well as the results obtained during the empirical evaluation. Section 5 summarizes the main results and concludes this work.

2 Related works and Problem Setting

This section introduces related works, formalizes the problem setting and presents the fundamental notions on which the models that will be introduced afterwards are based; notations are also defined hereafter.

Eliciting implicit evocations is closely related to well-known problems studied in Artificial Intelligence, in particular the reversed dictionary task, Topic Modeling, Language Model and text summarization. In the reversed dictionary or word access task, a word has to be found considering a given description; a problem closely related to the one considered in this paper - related recent work also refer to phrase embedding [6,15]; these approaches consider known

[2] We do not consider in this paper the complex problem of detecting implicit evocations. Note also the special syntax used to refer to the non-ambiguous entity reference `Banana` compared to its ambiguous lexical form *banana*.

term descriptions which is not considered hereafter. Topic Modeling techniques, for instance, can be used to analyze large corpora in order to generate topics by detecting frequent word collocations [14]. The aim of these approaches is slightly different since generated topics have to be extracted from large corpora by analyzing word usage statistics – topics are also *per definition* always abstract notions and cannot therefore be used straightforwardly for eliciting potentially specific entities. Considering our setting, it could be tempting for example to define a probabilistic model based on a conditional probability estimation enabling to compute $p(\texttt{Banana}|$*'yellow', 'tropical fruit', 'monkeys'*$)$. More generally, the problem could be studied by considering an approach based on language models - i.e. models largely used in machine translation, speech recognition or text summarization to cite a few. However, this study does not consider such models due to the curse of dimensionality [9] hampering their use for eliciting implicit evocations - indeed, despite the use of existing smoothing techniques [3], computing language models taking into account potentially large contexts (e.g. 5 to 10 words) is not possible. Other techniques based on neural probabilistic language models could also be considered to answer this limit [2]; more recent techniques based on sequence learning, e.g. based on Long Short-Term Memory neural network architectures, could also be worth studying [7]. Such techniques will only be partially and indirectly considered through the use of word embeddings techniques.

2.1 Explicit and Implicit Evocations

In this contribution we are interested by detecting implicit evocations; we introduce this notion by providing some illustrations as well as elements of definition – relationships with state-of-the-art notions such as topic identification or NER have been mentioned above.

First of all, entity evocations are here defined as strongly supported references to non-ambiguous notions or entities. As an example, several evocations could be detected from the following sentences "*I went to Paris last week, the Eiffel Tower is amazing... I love France!*". It is relatively easy to detect that it is highly probable that a reference to `Paris`, the capital of `France` called *Paris* is made. Note however that due to the ambiguous nature of words, it could not be the case; the word, i.e. surface form, *Paris* could indeed refer to other cities, e.g. `Paris_(Tennessee)`, or even other entities that are not locations. Nevertheless, considering the context which is defined by the sentence meaning, and in particular the utterrances of the words *France*, and *Eiffel Tower*, most people would understand the utterance of the string *Paris* as a will of the speaker to explicitly refer to the capital of France; here we consider that an *explicit* evocation has been made since a word corresponding to a lexical form of the entity, despite being ambiguous, explicitly refers to it. It is important to understand that evocations are here not necessarily understood as the intended speaker evocation; they are rather considered to be the consensual agreement towards understood evocations, i.e. the disambiguation most people would consider based on the context of utterance of words - e.g. as an example, nothing

restrict the speaker of aforementioned sentences to say that he is refering to `Paris_(Tennessee)`; even if most person would agree that discussing with such a speaker would thus be quite challenging. We therefore consider that, in most cases, intended evocations correspond to evocations most target recipients of a spoken/written message would consider. Explicit entity evocations could also be more refined than single word utterances, e.g *The City of Lights* could be used to mention `Paris`. All the examples provided so far were referring to the notion of explicit entity evocations since all of them could have been linked to a unique lexical/surface form of the entity.

Implicit entity evocations refer to entity evocations that cannot be directly associated to a word utterance, i.e. a surface form. As an example, the sentence "*Bob bought an expensive red sport car of a famous italian brand*" is most likely to refer to the fact that Bob bought a car from the italian car brand, `Ferrari` - otherwise stated, most of us would understand *Bob bought a* `Ferrari`. Additional examples are provided in the introduction section. Note that we could discuss in details the technical differences that we consider between surface forms of an entity and implicit references. Indeed, in some cases, judgement aiming at distinguishing if an evocation is explicit or implicit may depend on subjective evaluations. As an example, considering that *The City of Lights* is an explicit reference to the city `Paris` could be surprising considering that mentioning *the capital of France* would be considered as an implicit reference to the same city. We therefore stress that we consider explicit references to be lexical/surface forms of a concept. We thus consider that the utterance *'The City of Lights'*, contrary to the utterance *'the capital of France'*, is a lexical entry linked to the concept `Paris` in an index (e.g. a dictionary). Thus, considering that the lexical entry *'the capital of France'* is no linked to `Paris` in any index - no dictionary will give you such a lexical form -, more refined techniques have to be used to elicit the reference to Paris. As an example, this implicit evocation could be detected by taking advantage of a database or a knowledge base for answering the question *What's the capital of France?* - a process which is highly more complex than searching for a specific entry into a lookup table index to further resolve any ambiguity associated to word utterances.

Note that, independently to any context, implicit entity evocations can also be considered from a set of words (Table 1). In that case, the problem setting is close to a simplified form of the Pyramid game[3] (considering no interaction and no word ordering): a set of words is provided and a unique implicit evocation has to be provided by considering semantic relationships between the words. This is the setting we consider in this paper.

2.2 Problem Setting and Global Strategies Evaluated

Formal definition. Considering a vocabulary T and a set of entities $\mathcal{E}$ partially ordered into a taxonomy $O = (\preceq, \mathcal{E})$, we are looking for a function:

$$f : \mathcal{P}(T) \rightarrow \mathcal{E} \tag{1}$$

[3] `https://en.wikipedia.org/wiki/Pyramid_(game_show)`.

Given words	Expected Evocation
place, study, teacher	`School`
food, italy, round, tomato	`Pizza`
yellow, fruits, monkeys	`Banana`
city, UK, capital	`London`

Table 1: Examples of evaluation entries

The function f therefore aims at reducing a set of terms into a unique entity reference corresponding to the implicit mentioned entity. More generally, we are looking for a total order $\preceq_{\mathcal{E}}$ among the entities w.r.t. their relevancy for summarizing a given set of terms $T' \subset T$. To this aim we are looking for a scoring function evaluating the relevancy to associate a specific evocation to a given set of terms:

$$s : \mathcal{P}(T) \times \mathcal{E} \rightarrow \mathbb{R} \tag{2}$$

We will focus on the definition of the scoring function s in this paper. We therefore consider the following definitions: $f(T') := \arg\max_{e \in \mathcal{E}} s(T', e)$; the considered total order $\preceq_{\mathcal{E}}$ is thus defined such as $s(T', e') \leq s(T', e) \rightarrow e' \preceq_{\mathcal{E}} e$.

Evaluated strategies Different types of knowledge have to be taken into account for detecting implicit evocations. Only considering our simplified problem setting in which a set of words is evaluated, two types of information seems important for answering the task; (i) abstract restriction and enumerations, as well as (ii) salient properties definitions. Examples are provided:

- *Abstract restriction / enumeration* - need for a partial ordering of entities. An implicit evocation often refers to a general class to which the target implicit evocation refers to, e.g. *Paris* refers to a specific `Capital`, *expensive red sport car* refers to a `Car`. In those cases, it's important to know what are the instances of a specific class in order to be able to consider potentially relevant restrictions - i.e. group of entities in which candidates will be evaluated. In a similar manner, by mentioning *Krakatoa, Etna, Mont St. Helens* or *Eyjafjallajokull* the concept `Volcano` is clearly implicitly mentioned by providing explicit references of specific instances of volcano. Detecting such implicit evocations requires taking advantage of knowledge representations that will be used to identify a set of evocations referring to an abstract class.
- *Salient property* : Most of us would link the evocations {*green, monster, angry, muscle*} to the concept `Hulk`; this is because `Hulk` has a *green* skin, has a *muscular* type, and refers to a famous *angry monster*. Detecting mentions of such an implicit evocation requires linking provided evocations to salient properties of an entity of interest. To this aim an approach enabling to link properties values to specific entities has to be defined.

In this paper we consider that exhaustive formalized knowledge bases answering our needs, i.e. defining extensive properties values for a large number

of entities, are not available.[4] Indeed, despite the large efforts made for defining extensive knowledge bases [1,11], the properties to be analyzed for detecting implicit evocations are too broad, e.g. despite an URI exists for the concept `Hulk`, no property defines its skin color in DBpedia. We however consider that large text corpora are freely available (as it is the case today - Wikipedia for instance), and that it could be an interesting strategy to try mixing large scale text analysis (e.g. for capturing word relatedness enabling to detect a '*sort of*' link between the words *green* and *hulk*), as well as large taxonomical ordering of entity provided by existing knowledge bases.

As it has been mentioned in the previous section, implicit entity evocations are tightly linked to the notion of context. An evocation is indeed often explained by utterances of words that could be linked to entities that are members of the same conceptual neighborhood. As an example, the implicit evocation of `Ferrari` mentioned earlier could have been explained by its narrow relationships with the concepts `car`, `Italy` and `brand`. Interestingly, the strength of a relationship between words or entities can therefore be discussed through the notions of semantic similarities/proximities [5].

In this context, we therefore propose to define and to compare different strategies taking advantage of (i) terms relationships extracted from large corpora analysis - through term semantic relatedness estimations -, as well as (ii) conceptual relationships defined by a partial ordering of entities provided by a knowledge base. The models discussed in this paper consider this postulate. Considering the type of strategies we will evaluate, two notions are of major importance: semantic relatedness of terms and semantic similarity of concepts/entities; both are briefly introduced in the following subsection.

2.3 Estimating similarities and relatedness of words and entities

Both word relatedness/proximity and entity similarity estimations from text and knowledge base analysis have been, and are still, extensively studied in particular by the NLP community. Word relatedness and entity similarity are extensively used in information retrieval, question answering, among others. A short introduction to these notions is provided hereafter - the reader can refer to the extensive literature and surveys for additional information, e.g. [5].

Estimating Word Relatedness Considering a vocabulary T, word relatedness estimations aim at defining a function $\sigma_{TT} : T \times T \rightarrow [0, 1]$ such as σ_{TT} enables capturing the intuitive (but weakly defined) notion of relatedness – generally defined as the strength of the semantic link established between units of language, here a pair of words [5]; once again most people will agree that the two words (*banana, monkey*) are more related than the two words (*banana, lion*).

Among the various approaches defined for comparing a pair of words, most recent strategies aims at (i) building a vector representation of words (called

[4] and that expecting such bases to exist in the near future is just illusionary.

embeddings) that will further be compared using traditional vector comparison metrics, most often the cosine similarity of vector representations. Technical details of most approaches therefore rely on defining the strategy used for building embeddings. Those strategies rely on the consideration that word meaning is defined by its context of use. Embeddings will thus be built by (indirectly) analyzing word collocations. Most recent strategies rely on predictive approaches, e.g. by building word embeddings by using internal representations of words that have been built by a neural network trained to predict a word considering a given context or a context considering given words. Further details related to word embeddings are out of the scope of this paper.

Estimating Entity Similarity Considering a partial ordering $O = (\preceq, \mathcal{E})$ among a set of entities $\mathcal{E}$ (individuals and concepts of a knowledge base). The similarity of two entities is defined by $\sigma_{\mathcal{E}\mathcal{E}} : \mathcal{E} \times \mathcal{E} \rightarrow [0, 1]$. An example of similarity measure proposed by Lin's measure is presented [8]:

$$sim(e, e') = \frac{2 \cdot IC(MICA(e, e'))}{IC(e) \times IC(e')} \tag{3}$$

with $MICA(e, e')$ the Most Informative Common Ancestor of entities e and e' with regards to a function evaluating the information content of an entity, with $IC : \mathcal{E} \rightarrow [0, 1]$, and $x \preceq y \rightarrow IC(x) \geq IC(y)$, i.e. an entity is always considered to be more informative than its ancestors, e.g. $IC(\texttt{Paris}) > IC(\texttt{Capital})$.

3 Models for Detecting Implicit Entity Evocations

This section presents the various model proposals that are used to distinguish a ranked list of entity evocations for a provided set of terms (by defining Equation 2, page 4). These models consider that word vector representations, as well as a labeling function linking terms to entities are provided. The labeling function defines the sets of labels that refer to a specific entity, i.e. $\pi : \mathcal{E} \rightarrow \mathcal{P}(T)$, e.g. $\pi(\texttt{Person}) = \{person, human, \ldots\}$.[5]
Two general types of models are presented:[6]

1. *Vector Aggregation Model* (VAM). The aim is to encompass the meaning of a set of terms by aggregating commonly used word embeddings.
2. *Graph-based Model* (GM). The aim is to detect implicit entity evocation by using a pre-built structure mixing both links between entities and terms as well as relationships between terms.

[5] Note the ambiguity at terminological level, a given term can refer to several entity. In addition, due to the transitivity induced by the relationship defining the considered partial ordering O, the set of entities that are potentially, implicitly or explicitly, evocated by a term $t \in T$ is defined by the set: $\bigcup_{e \in \mathcal{E}, t \in \pi(e)} \{x | e \preceq x\} \subseteq \mathcal{E}$; considering $\texttt{car} \prec \texttt{vehicule}$, mentioning `car` makes you implicitly mention `vehicule`.

[6] Nothing excludes that specific models generated by one approach cannot be expressed by the other approach.

Both models are detailed hereafter. They will next be compared by analyzing their performances w.r.t an empirical evaluation.

3.1 Vector Aggregation Model

The Vector Aggregation Model (VAM) relies on a generic three-step strategy for analyzing a set of terms $T' \subset T$:

1. Computation of a conceptual evocation vector $\boldsymbol{t} \in \mathbb{R}^{|\mathcal{E}|}$ for each term $t \in T'$ - the aim of this representation is to encompass all potential explicit and implicit entity evocations that are made by t.
2. Aggregation of the entity evocation vector of the terms composing the set of terms to evaluate. We will evaluate aggregations that generate vector representations of T' into $\mathbb{R}^{|\mathcal{E}|}$.
3. Analysis of aforementioned aggregation product in order to compute the ranked list of entity evocations.

These three steps are detailed.

Establishing the link between words and entities. We consider that without prior knowledge about context, the degree of evocation of an entity by a term is defined by the function $\sigma_{T\mathcal{E}} : T \times \mathcal{E} \rightarrow [0, 1]$, defined such as:

$$\sigma_{T\mathcal{E}}(t, e) = \max_{t' \in \pi(e)} \sigma_{TT}(t, t') \tag{4}$$

Otherwise stated, the relationship considered between a term and an entity only depends on the semantic relatedness that can be distinguished at word level. Note that no prior knowledge about word usage is taken into account in this approach. Therefore, considering a term t, every entity $e \in \mathcal{E}$ with $t \in \pi(e)$ will have the same $\sigma_{T\mathcal{E}}(t, e)$ value – which will be maximal if the σ_{TT} function respects the identity of the indiscernible.[7]

Finally, without applying any preprocessing step excluding potential conflicting entity evocations, we consider the evocation of a term $t \in T$ to be defined by the function $\rho_{T\mathcal{E}} : T \rightarrow \mathbb{R}^n$ with ($|\mathcal{E}| = n$):

$$\rho_{T\mathcal{E}}(t) = [\sigma_{T\mathcal{E}}(t, e_1), \ldots, \sigma_{T\mathcal{E}}(t, e_n)]^\intercal \tag{5}$$

[7] Otherwise stated, by observing the word utterance *Paris*, all concepts having this specific string as label, e.g. `Paris` (France), `Paris_Tennesse`, will have the same evocation degree value. This is obviously not how humans process information. Indeed, without context, or only considering poor contextual information, people rely most often on evocation likelihood (considering their body of knowledge). Therefore, to refine the approach, we could also estimate the probability that a given term refers to an entity. Several approaches could be explored, e.g. analyzing usage of WordNet synsets. This information is however difficult to obtain for entities that are not mentioned into this structured lexicon, which hampers the general aspect of the approach. We therefore consider that no prior knowledge about word-entity evocation is provided by excluding the use of statistics about word-entity usage.

This vector represents the potential entity evocations of a term without distinguishing among potentially conflictual evocations. We however consider that it represents a footprint encompassing all entity evocations a word could refer to.

Aggregation of the information provided by several words. Several approaches can be considered for aggregating the degrees of evocation of a set of terms $T' \subset T$. To this purpose we consider a general function $P_{\mathcal{E}} : \mathcal{P}(T) \rightarrow \mathbb{R}^n$. Two definitions of $P_{\mathcal{E}}$ will further be considered; both of them are based on an element-wise aggregation: (i) $P_{\mathcal{E}}^{min}(T') = \wedge_{t \in T'} \rho_{T\mathcal{E}}(t)$ defining the aggregation to be the minimal evocation value among all terms, and (ii) a less constraining evaluation summing the evocations $P_{\mathcal{E}}^{sum}(T') = \sum_{t \in T'} \rho_{T\mathcal{E}}(t)$.

Ranking conceptual evocations. We consider that, because of the nature of the function used to build the vector representations, as well as the aggregation operator, implicitly mentioned entities could be detected by analyzing associated dimension values in $P_{\mathcal{E}}$. More precisely, it is expected that evocation values associated to implicitly mentioned entities will diverge from the values that would be expected if randomly selected terms were used to build the vector representation. We therefore consider that the distribution of the value for a given entity and a given size of set of terms is known. This distribution is estimated by computing associated $P_{\mathcal{E}}$ representations for randomly sampled sets of terms of a specific size. The distribution stores for each entity the number of time a randomly composed set of terms has obtained a specific evocation value. Using this estimated distribution we can compute the probability that the observed value for a given set of terms is an *artefact*, or indeed seems to refer to an implicit evocation. We therefore consider that implicitly evocations are those for which observed values highly diverge from the expected one.
Several approaches have been tested for defining the ranking function; the raw score (a metric taking on the standard deviation[8] σ and the mean μ) is presented. Considering a given set of term T', we denote rs_{e_i} the raw score of T' w.r.t $e_i \in \mathcal{E}$:

$$rs_{e_i}(T') = \frac{P_{\mathcal{E}}(T')_i - \mu_{e_i}}{\sigma_{e_i}} \tag{6}$$

μ_{e_i} and σ_{e_i} respectively denote the median and the standard deviation of evocation values for the entity e_i computed during the sampling process associated to samplings of size $|T'|$.

3.2 Graph-based Model

The Graph-based Model (GM) approach is based on a graph propagation strategy aiming at distinguishing what are the most relevant entities to be considered given a set of terms. Defined graph data structure aims at modelling relationships: among the terms, among the entities, as well as among terms and entities.

[8] Recall standard deviation: $\sigma = \sqrt{E[X^2] - E[X]^2}$.

We first present the graph structure. Next, the propagation approach used for distinguishing entity evocations is introduced.

Graph model. Formally, let's consider a weighted directed graph $G = (V, E)$ with $V = T \cup \mathcal{E}$ and $E \subseteq V \times V$. Three types of relationships are distinguished:

1. relationships among terms, i.e. from $T \times T$; those relationships are weighted using a σ_{TT} measure capturing the relationships among terms. The weight of a relationship $(t, t') \in E$ is defined by a function $w_{TT} : T \times T \to [0, 1]$:

$$w_{TT}(t, t') = \frac{\sigma_{TT}(t, t')}{\sum_{t'' \in T} \sigma_{TT}(t, t'')} \tag{7}$$

   This weighting function definition aims at normalizing the σ_{TT} scores considering that scores distributions may highly differ between terms.

2. relationships among entities, i.e. from $\mathcal{E} \times \mathcal{E}$; those relationships are given by the partial ordering O; the weight of the relationships are provided by a $\sigma_{\mathcal{E}\mathcal{E}}$ measure. More precisely, the relationships between entities are defined as follows: (1) building of a graph $G' = (\mathcal{E}, E_{\mathcal{E}\mathcal{E}})$ from O by considering that $(e, e') \in E_{\mathcal{E}\mathcal{E}}$ iff $e \preceq e'$ or $e' \preceq e$ in O; (2) apply a transitive reduction to G';(3) weigh the relationships considering a $\sigma_{\mathcal{E}\mathcal{E}}$ measure – the weights are here also defined by normalizing considering all relationships defined in G'.

$$w_{\mathcal{E}\mathcal{E}}(e, e') = \frac{\sigma_{\mathcal{E}\mathcal{E}}(e, e')}{\sum_{e'' \in \mathcal{E} | (e, e'') \in E_{\mathcal{E}\mathcal{E}}} \sigma_{\mathcal{E}\mathcal{E}}(e, e'')} \tag{8}$$

3. relationships between terms and entities, i.e. from $(T \times \mathcal{E}) \cup (\mathcal{E} \times T)$; those relationships are given by the labeling function π. With $e \in \mathcal{E}, t \in T$, we consider that both $(t, e) \in E$ and $(e, t) \in E$ iff $t \in \pi(e)$, i.e. iff the term t is a label (refers) to the entity e.[9]

Propagation Model . Considering a given set of terms $T' \subset T$. The propagation model adopted to distinguish relevant entities is defined in Algorithm 1; the propagation procedure is detailed by Algorithm 2. The proposed approach is discussed hereafter. As it is defined in Algorithm 1, the global strategy aims at:

1. Computing the entity evocation degree for each term composing the query (lines 3-9). This is done by propagating a fixed quantity from each node composing the query (line 7).
2. Aggregating those results in order to compute, for the full set of terms, the entity evocation scores for each entity (lines 10-13).

[9] Those relationships could have also been weighted by considering word usage frequency. However, as stated before, we consider that no weighting function is defined here - even if analyzing σ_{TT} scores distributions could have been used.

The details of Algorithm 1 are now provided. At line 1-2, we initialize the map data structures.[10] that will be used to store the (temporary) results. The entity evocation degrees for each query term is stored into *query_term_evocation_map* – for instance, the entity evocation for the term t is stored as a map into *query_term_evocation_map*$[t]$; *query_term_evocation_map*$[t][e]$ is the evocation degree of entity e by the term t. From line 3 to 9 we compute the entity evocations for each term defining the query (discussed later). From lines 10 to 13 those results are aggregated using a specific strategy. The sum and the median will be considered - intuitively, the median is used to express the fact that we not only want a high score; but we also want the score to be supported by a shared contribution of the terms composing the query.

Algorithm 1: Propagation algorithm

Data: The graph G structuring terms and entities; a set of terms $T' \subset T$, with $|T'| << |T|$, ϵ threshold value: stopping criteria.
Result: A data structure storing the relevance of each entity.

```
1  query_term_evocation_map ← map()
2  concept_score ← map() ;
3  for t ∈ T' do
4  |  ev_map ← map() ;
5  |  visited_node ← {} ;
6  |  score ← 1;
7  |  propagate(t, visited_node, 1, ev_map) // cf. Algorithm 2;
8  |  query_term_evocation_map[t] = ev_map ;
9  end
10 for e ∈ E do
11 |  entity_scores[e] = aggregate(e, query_term_evocation_map);
12 |  // The aggregate function can just be a sum, min, average...;
13 end
14 return entity_scores;
```

Details of the propagation are defined by Algorithm 2.[11] The propagating process is defined using a recursive procedure aiming at propagating values avoiding already processed nodes. Depending of the type of node being processed (term or entity), the propagation aims at extending to other terms or entities. When a term is processed (line 2 to 13) a quantity is propagated to all entities that could be referred by the term (without any *a priori* consideration about term usage). The evocation is next propagated to those entities if the propagated quantity is important enough (line 6). The propagation is also performed to the neighboring terms by taking into account the distance between the terms at terminological level – line 8 to 10. When an entity node is processed (line 13 to 22) the propagation to the terminological level is performed by considering the labels associated to the entity. The propagation is also performed at the entity level, also taking into account the entity similarity that can be computed by analyzing entities' topological ordering (cf. weight definition Equation 8).

[10] A map or dictionary stores a value for a specific key.

[11] We consider to be known G, T and $\mathcal{E}$ the terms and entities, ϵ the threshold value defining when to stop the propagation, $synDecFactor$ a decay factor for handling synonyms while propagating, $eSmoothingFactor$ a smoothing factor for reducing the impact of excessively considering the taxonomy on the results.

Algorithm 2: *propagate* routine

Data: A given node v of the graph, a set of visited node S, *score*: a score value (to propagate), *qtem* (for *query_term_evocation_map*): a map for storing entity evocation scores.
Result: None - updated evocation vector

$S.add(v)$
if $v \in T$ **then**
  $\mathcal{E}' = \{e \in \mathcal{E} | v \in \pi(e)\}$
  for $e \in \mathcal{E}'$ **do**
    $qtem[e] = qtem[e] + score$
    if $score \geq \epsilon$ *and* $e \notin S$ **then** $propagate(e, score)$;
  end
  for $t \in T$ **do**
    $p\_value \leftarrow w_{TT}(t, v) \times score$
    if $p\_value \geq \epsilon$ *and* $t \notin S$ **then** $propagate(t, p\_value \times synDecFactor)$;
  end
end
else
  // $v \in \mathcal{E}$
  for $t \in \pi(v)$ **do**
    if $score \geq \epsilon$ *and* $t \notin S$ **then** $propagate(t, p\_value)$;
  end
  for $e \in \{e \in \mathcal{E} | (v, e) \in E \vee (e, v) \in E\}$ **do**
    $p\_value \leftarrow w_{\mathcal{E}\mathcal{E}}(v, e) \times score \times eSmoothingFactor$
    if $p\_value \geq \epsilon$ *and* $e \notin S$ **then** $propagate(e, p\_value)$;
  end
end
$S.remove(v)$

4 Evaluation and Results

4.1 Evaluation Protocol

The proposed evaluation is based on a set of expected entity evocations for given sets of words. Table 1 presents some of the 220 entries composing the evaluation set. Expected implicit evocations for each entry have been linked to WordNet 3.1 [11], a widely used lexical database. WordNet defines an ordering among sets of synonyms providing both, (i) the set of entities and their partial ordering (O), as well as (ii) the labeling function - π function.

The performance of the different approaches is evaluated by considering the number of queries for which the expected answer is provided among the top k results. For each approach, six evaluation settings have been compared by evaluating if the expected answer is found among sets composed of 1, 2, 3, 5, 10 or 20 best ranked results. In each setting, the ranked list of entities is computed by considering a set $\mathcal{E}' \subset \mathcal{E}$ corresponding to the expected answers for all evaluated queries ($|\mathcal{E}'| = 198$). Implementations of the $\sigma_{\mathcal{E}\mathcal{E}}$ measure have been made using SML (Semantic Measures Library)[4]. The σ_{TT} function used in the experiments uses Glove word embeddings [13]. Datasets, tested methods Java implementations as well as complete technical details about the evaluation are provided at `https://github.com/sharispe/ICE`.

Six models have been evaluated:

- Two Vector-based Aggregation Model definitions: VAM_MIN uses an aggregation strategy based on the min, VAM_SUM uses the sum.
- Four Graph-based Model (GM) definitions: two strategies using an aggregation approach based on median, using propagations at entity level or not (GM_MEDIAN_KB and GM_MEDIAN respectively); two strategies using an aggregation approach based on sum, using propagations at entity level or not (GM_SUM_KB and GM_SUM respectively).

4.2 Evaluation Results

Approach k=	1	2	5	10	20	50	100
VAM_MIN	0.17	0.25	0.34	0.41	0.51	0.67	0.81
VAM_SUM	**0.28**	**0.39**	0.51	0.6	0.66	0.8	0.86
GM_MEDIAN	0.20	0.29	0.46	0.54	0.59	0.66	0.74
GM_SUM	0.18	0.29	0.47	0.59	0.65	0.84	0.88
GM_MEDIAN_KB	0.24	0.35	0.49	0.58	0.64	0.74	0.81
GM_SUM_KB	0.23	0.32	**0.55**	**0.63**	**0.72**	**0.86**	**0.9**

Table 2: Evaluation results (recall).

Results are presented in Table 2. Considering the performance of evaluated systems setting k to 1 and 2, the results show that the best performance is obtained using a Vector-based Aggregation Model configuration taking advantage of the sum aggregation approach (VAM_SUM). It is interesting to underline that this approach does not take into account any information provided by the ordering of entities - while providing a 0.05 recall improvement over the best results that have been obtained using an approach taking advantage of taxonomic information (GM_MEDIAN_KB). Note also the critical impact of modifying the aggregation strategy using a VAM approach: by using a min aggregation strategy the performance highly decreases (e.g. a 0.11 difference is observed between VAM_MIN and VAM_SUM using k=1). Considering the graph-based approach, the results highlight a large benefit of using taxonomical information for eliciting implicit entity evocations. Indeed, using both median and sum approaches, incorporating information provided by the taxonomy leads to a significant performance increase (cf. comparison of the scores between GM_MEDIAN/GM_MEDIAN_KB, as well as GM_SUM/GM_SUM_KB). It is finally worth noting that by setting k greater than 2, the best performances are achieved using a graph-based model taking advantage of taxonomical information. These results stress that using taxonomical information helps better identifying the semantic neighborhood of expected results, e.g. setting k=20 GM_SUM_KB achieves a 0.72 recall while the VAM_SUM performance is 0.66.

5 Conclusion

In this paper, we have introduced the challenge of eliciting implicit entity evocations by stressing (i) its applications for improving automatic approaches en-

abling human spoken and written productions to be deeply understood, and (ii) its link to existing NLP and AI challenges (e.g. NER, Topic Modelling, Language Model). Several models mixing word embeddings analysis and symbolic representations provided by existing knowledge bases have been proposed. These models can be used to distinguish relevant implicit entities mentioned from a set of terms - they can therefore be used as core elements of more complex systems aiming at providing automatic analysis of the semantics of large units of language. The preliminaries results obtained in the performed experiments highlight the potential benefits of defining an hybrid approach combining word embeddings with symbolic representations for the task - even if additional experiments and configuration settings have further to be proposed and evaluated. To this aim, implementation source code, evaluation dataset and details of the performed experiments are shared to the community (cf. Section 4).

References

1. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: Dbpedia: A nucleus for a web of open data. The semantic web pp. 722–735 (2007)
2. Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C.: A neural probabilistic language model. Journal of machine learning research 3(Feb), 1137–1155 (2003)
3. Chen, S.F., Goodman, J.: An empirical study of smoothing techniques for language modeling. In: Proceedings of the 34th annual meeting on Association for Computational Linguistics. pp. 310–318. Association for Computational Linguistics (1996)
4. Harispe, S., Ranwez, S., Janaqi, S., Montmain, J.: The Semantic Measures Library and Toolkit: fast computation of semantic similarity and relatedness using biomedical ontologies. Bioinformatics 30(5), 740–742 (2014)
5. Harispe, S., Ranwez, S., Janaqi, S., Montmain, J.: Semantic Similarity from Natural Language and Ontology Analysis, vol. 8. Morgan & Claypool Publishers (2015)
6. Hill, F., Cho, K., Korhonen, A., Bengio, Y.: Learning to understand phrases by embedding the dictionary. Transactions of the Association for Computational Linguistics (2015)
7. Jozefowicz, R., Vinyals, O., Schuster, M., Shazeer, N., Wu, Y.: Exploring the limits of language modeling. arXiv preprint arXiv:1602.02410 (2016)
8. Lin, D., et al.: An information-theoretic definition of similarity. In: ICML. vol. 98, pp. 296–304 (1998)
9. Manning, C.D., Raghavan, P., Schütze, H., et al.: Introduction to information retrieval, vol. 1. Cambridge university press Cambridge (2008)
10. Mendes, P.N., Jakob, M., García-Silva, A., Bizer, C.: Dbpedia spotlight: shedding light on the web of documents. In: Proceedings of the 7th international conference on semantic systems. pp. 1–8. ACM (2011)
11. Miller, G.A.: Wordnet: a lexical database for english. Communications of the ACM 38(11), 39–41 (1995)
12. Nadeau, D., Sekine, S.: A survey of named entity recognition and classification. Lingvisticae Investigationes 30(1), 3–26 (2007)
13. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: EMNLP. vol. 14, pp. 1532–1543 (2014)
14. Steyvers, M., Griffiths, T.: Probabilistic topic models. Handbook of latent semantic analysis 427(7), 424–440 (2007)
15. Thorat, S., Choudhari, V.: Implementing a reverse dictionary, based on word definitions, using a node-graph architecture. Proceedings of COLING 2016 (2016)