Definition Frames: Using Definitions for Hybrid Concept Representations

Evangelia Spiliopoulou
School of Computer Science
Carnegie Mellon University
espilio@cs.cmu.edu

Eduard Hovy
School of Computer Science
Carnegie Mellon University
hovy@cmu.edu

Abstract
Concept representations is a particularly active area in NLP. Although recent advances in distributional semantics have shown tremendous improvements in performance, they still lack semantic interpretability. In this paper, we introduce a novel hybrid representation called Definition Frames, which is extracted from definitions under the formulation of domain-transfer Relation Extraction. Definition Frames are easily reformulated to a matrix representation where each row is semantically meaningful. This results in a fluid representation, where we can prune dimension(s) according to the type of information we want to retain for any specific task. Our results show that Definition Frames (1) maintain the significant semantic information of the original definition (human evaluation) and (2) have competitive performance with other distributional semantic approaches on word similarity tasks. Furthermore, our experiments show substantial improvements over word-embeddings when fine-tuned to a task even using only a linear transform.

1 Introduction
Creating algorithms that extract human knowledge is a fundamental question for Natural Language Processing. An important problem in this area is how to construct concept representations and determine which information to encode.

Ontology-based methods constitute one of the oldest approaches to organize and represent knowledge that is still widely used in NLP tasks. They can be in the form of lexical resources like WordNet (Miller 1995) and FrameNet (Baker, Fillmore, and Lowe 1998), large knowledge bases like ConceptNet (Speer and Havasi 2012) and CYC (Lenat 1995) or domain-dependent ontologies carefully designed for particular problems/domains. Ontologies are particularly useful since they contain accurate and semantically interpretable information that can be easily accessed and filtered by humans according to the task of interest. However, this information is typically constructed manually, which is a very time-consuming and difficult process. This results in representations that are not easily extensible, so they cannot be modified or fine-tuned in the presence of new information.

Most recent advances focus on learning representations by training a language model on extremely large corpora. Although this is a more data-driven approach compared to the meticulous construction of an ontology, distributed representations are fully automated (no manual annotations needed) and they can be fine-tuned for any new task. Earlier examples include models like GloVe (Pennington, Socher, and Manning 2014), word2vec (Mikolov et al. 2013) and fastText (Bojanowski et al. 2017), which can be used to obtain high quality generic word embeddings by pre-training them in large corpora. Most recent work focuses on context-sensitive word embeddings like ELMo (Peters et al. 2018) and BERT (Devlin et al. 2018), which achieve significant improvements in various downstream NLP tasks. Those methods can represent polysemy, since the word embeddings are no longer static, but they change based on the context that the word occurs.

Despite their exceptional performance, most distributional methods do not have any explicit semantic interpretation. The resulting representations may encode tremendous amount of information, but we have no control or way to interpret what this information is, how it relates to the concept or if it just reflects biases of the data. Thus, we cannot choose which type of information is useful for a specific task, unless we have a lot of data and resources to fine-tune the representations (which, unfortunately, is a rare scenario for most semantically-oriented tasks). Although few approaches have tried to bridge the gap between semantics and distributed representations (Faruqui et al. 2015; Mrkšić et al. 2017), they only encode information from manually constructed ontologies. This causes serious limitations since most available information is either noisy or in a free text format. Furthermore, although those approaches use ontological relations, the resulting representation is a word embedding without any further semantic interpretation.

Motivated by these problems, we introduce a novel hybrid representation called Definition Frames that encodes semantic information extracted from definitions. This information is extracted automatically via a relation extraction model, which means that we can create a representation for any term, as long as there is some accompanying definition/text. According to our knowledge, Definition

1Code will be released upon publication
Frames are the first hybrid representation: they have an explicit structure due to their semantically meaningful rows, while maintaining the properties of distributional semantics. As our experiments show, Definition Frames achieve better performance in word similarity tasks, when used as a post-processing method.

2 Prior Work

Dictionary definitions constitute an excellent source of human knowledge, as they contain essential relations about a concept. Although definitions are written in natural language, they follow a specific structure. Most definitions of a concept contain the class to which it belongs (Genus) and the properties that differentiate it from other concepts of the same class (Differentia). In addition to their structure, definitions contain generic information that is sufficient to uniquely identify a concept, whereas most natural language text (i.e. news articles, books, online forums) typically contain information about specific instances of a concept. Those interesting properties of definitions motivate a series of work that uses them as sources to extract knowledge.

Earlier work on definitions focuses on extracting the Genus and Differentia relations via string matching heuristics and syntactic properties (Binot and Jensen 1993; Calzolari 1984; Chodorow, Byrd, and Heidorn 1985). However, similarly to ontology-based representations, those methods require a lot of manual effort and lack generalization. Recent approaches on information extraction from definitions try to directly encode definitions to distributed representations. The motivation behind this work is to benefit from the rich knowledge encoded in definitions, while still maintaining the properties of distributional semantics. Tissier, Gravier, and Habrard use a skip-gram model to obtain word embeddings trained on dictionary definitions. Inspired by the work of Noraset et al. on generating definitions from word embeddings, Bosc and Vincent use an auto-encoder on definition sentences, whereas the hidden layer is used as the distributed representation. Other work includes binary classification of sentences to definitional or not (Anke and Schockaert 2018) and reverse dictionary look-up (Hill et al. 2016; Zock and Bilac 2004).

Another line of work focuses on enriching word embeddings with semantic knowledge from lexical resources, typically in a post-processing manner. Faruqui et al. propose Retrofitting, a process where they use belief-propagation to update embeddings on a relation graph from a large ontology. Mrkšić et al. and Mrkšić et al. on the other hand, inject antonymy and synonymy constraints into word embeddings, a process they call Counter-fitting. An interesting example of Counter-fitting is the LEAR framework, where they discuss the particular importance of the isA relation in word embeddings (Vulić and Mrkšić 2018).

3 Approach

3.1 Definitional Relations

Similar to work on definitions, relation extraction focuses on detecting a set of important relations between terms. Besides domain-specific relations, most RE tasks (Gábor et al. 2018; Hendrickx et al. 2009) typically contain relations that belong to three main classes: hypernymy/hyponymy relations (isA), relations about structure (madeOf, partOf, hasA) and teleological relations (usedFor, cause). In order to verify the prevalence of those relations in definitions and their correspondence to Genus and Differentia, we manually annotate 50 sentences defining a concept chosen at random (summarized in Table 1). Those concepts are selected from the set of all nominal synsets from WordNet that are linked to Wikipedia, while for the sentences/definitions we use the first sentence of Wikipedia. From those annotations we observed that most definitions use the isA relation combined with a Differentia type relation and that certain relations can only be used on concepts with specific semantic types. As an example, the cause relation can only be used on events, while the madeOf relation on physical entities. Some other structures used as Differentia include adjectives, topics and analytical descriptions of processes.

| Relation | Num Sentences |
|----------|---------------|
| IsA      | 44            |
| PartOf   | 7             |
| HasA     | 9             |
| MadeOf   | 2             |
| UsedFor  | 10            |
| Cause    | 4             |

Table 1: Annotated Relations for 50 Wikipedia Sentences.

3.2 Data Construction

Because there is no prior work on neural-based RE from definitions, we follow a domain adaptation technique where we use a model pre-trained on different data. However, most existing datasets on RE are particularly small or focus on a very narrow domain, which makes it hard to use them to obtain general relations. Given those constraints, we construct a large but simple dataset based on ConceptNet to pre-train the Relation Retriever model (more details in section 3.3).

ConceptNet (Speer and Havasi 2012) is a large general purpose ontology that contains relations between pairs of concepts. Many of those relations are accompanied by a small source-definition, where the relation was extracted from. For example, in Figure 1 we see that the Concept-query Sun is linked to two sentences (Sun is a star and Sun is in our Solar System) from ConceptNet with the corresponding Definitional Relations isA and partOf. In order to construct the training data, we first extract all ConceptNet relations that overlap with Definitional Relations (isA, usedFor, partOf, hasA and madeOf). Then, for each pair of concepts, we extract the POS and chunk tags using the Stanford CoreNLP parser (Manning et al. 2014). We also mark the concept that corresponds to the first argument of the relation, as it represents the term for which we want to extract the Definition Frames given its definition (Concept-query).}

\footnote{The hasA relation is the inverse of partOf.}

\footnote{We exclude the cause relation, as our evaluation datasets typically do not include events.}
Those are used as additional features to the initial sentence in the Relation Retriever model. In order to select the best performing model, we split our data into train (68,700 relations), dev and test (8,500 relations respectively).

In order to extract the Definition Frames we use data from Wikipedia which we pre-process in a similar way. One major difference compared to ConceptNet is that Wikipedia sentences are more complex, as they may contain relations of the Concept-query with multiple terms or even relations between terms other than the Concept-query. In order to account for those differences, we do not add any constraints on the number of the extracted relations.

### 3.3 Extracting Definition Frames

Our framework consists of two parts: the Relation Retriever and the Definition Encoder. Given a Concept-query, the Relation Retriever uses the corresponding Wikipedia sentence to extract the terms that are related to that concept. The set of the extracted relations with the respective related terms form the Definition Frame.

As an example, consider the Concept-query Moon for which we want to extract the Definition Frames. As we see in Figure 1, we first extract the Wikipedia definition about Moon. This sentence is then processed in the pre-trained Relation Retriever model, which detects the terms that are related to Moon. In our example those terms are satellite, astronomical body and Solar System. Those terms with their corresponding relations constitute the Definition Frame for Moon.

Since our setting is different from typical relation extraction tasks and ConceptNet data is fairly simple compared to Wikipedia definitions, we choose to avoid over-complicated models for the Relation Retriever, as they are prone to overfitting. Thus, for our model selection we perform experiments with models that in general constitute strong baselines for RE tasks and do not take into account specific properties of the data. Those models include: a simple BiLSTM (Hochreiter and Schmidhuber 1997), a 2-layer deep BiLSTM (Stacked-BiLSTM) and a hybrid BiLSTM-character-CNN model that shows high performance on NER tasks (Ma and Hovy 2016). Although our goal is not to detect named entities, NER is a problem highly correlated with our setting, since we do not have gold entities (besides the Concept-query).

As we see in Table 2 all models have extremely good performance, which is probably due to the simplicity of the ConceptNet dataset. Given that the simple BiLSTM shows slightly better results while having the smallest number of parameters, we select it as the main model in the Relation Retriever module.

### 3.4 Encoding Definition Frames

In the previous section we described how we obtained the Definition Frames for a Concept-query. Although Definition Frames capture important information to define a concept, we still face the problem of how to use them in a downstream NLP task. In this section we explain our method to
encode them in a distributed representation via the Definition Encoder.

The output representation from the Definition Encoder is a matrix where each row corresponds to one of the Defi-
nitional Relations. Given a relation \( r_i \), the corresponding \( i \)th row of the matrix is an encoding of the terms related to
the Concept-query with the same relation \( r_i \), as provided by
the corresponding Definition Frame. The Definition Encoder
uses an embedding space (we refer to this as \( \text{Basis} \)) to con-
struct the individual word embeddings for the related terms.

Specifically, given a Definition Frame \( F = \{ r_1 : S_1, \ r_2 : S_2, \ldots, r_k : S_k \} \), where each \( r_i \in \{ \text{isA}, \text{usedFor}, \text{partOf}, \text{hasA}, \text{madeOf}, \text{cause} \} \) and \( S_i \) is the set of terms related
to the Concept-query with the relation \( r_i \), we define the
average embedding \( w_i \) for relation \( r_i \) as:

\[
w_i = \frac{1}{|S|} \sum_{s \in S_i} \text{Basis}(s)
\]

where \( \text{Basis}(s) \) is the embedding for each word \( s \) based
on the input Basis space. Then, we construct the matrix \( DF \),
where each dimension \( i \) contains the vector \( w_i \) and seman-
tically corresponds to the terms that relate to the Concept-
query via the relation \( r_i \). All encoded Definition Frames
maintain the same structure (each row corresponds to a fixed
relation), thus a semantically meaningful representation. If
no terms were extracted for a relation, we use the zero vec-
tor of the appropriate size instead of \( w_i \). An example of the
encoded Definition Frame for the concept \textit{Moon} is shown in
Figure 1, where each dimension corresponds to a unique rel-
tion (\textit{isA} and \textit{partOf} relations have encoded embeddings,
while the others correspond to zero vectors).

### 4 Experiments & Discussion

#### 4.1 Evaluation on Word-Similarity Tasks

This set of experiments focuses on the performance of Defi-
nition Frames on word similarity tasks and how we can ben-
efit from their inherent structure. Our experiments are based
on benchmark word-similarity datasets and code, as pro-
vided by Faruqui and Dyer, for which we report Spearman’s
correlation \( \rho \) between the cosine similarity of the words
representations and the normalized ground truth similarity
score. For all experiments we only consider words that exist
in all our compared methods and baselines.

Word-similarity tasks are particularly interesting, as
words can be similar in different ways or facets. Although
most of our data does not have an explicit type of similar-
ity, we can divide them into two broad categories, as prior
literature suggests: similarity and relatedness. For similar-
dity datasets we use RG-65 (Rubenstein and Goodenough
1965), SimLex999 (Hill, Reichart, and Korhonen 2015),
SimVerb3500 (Gerz et al. 2016) and MC-30 (Miller and
Charles 1991), while for relatedness we use MEN (Bruni
et al. 2012), MTurk287 (Radinsky et al. 2011), MTurk771
(Halawi et al. 2012) and RW-Stanford (Luong, Socher,
and Manning 2013). Furthermore, we evaluate on WS-353
dataset (Finkelstein et al. 2002) by dividing it into similarity
and relatedness subsets (WS-SIM and WS-REL), as pro-
scribed by Agirre et al.

#### The Role of Structure

We perform experiments with three different types of word embeddings that vary with respect to the method and the data they were trained on. Those include: GloVe embeddings pretrained on Wikipedia (directly provided from Pennington, Socher, and Manning 2014)), word2vec trained on WordNet definitions (as described in Bosc and Vincent) and dict2vec trained on Wikipedia (using the code available from Tissier, Gravier, and Habrard). Given that dict2vec is also a post-processing method on word2vec via definitions, we are not comparing with additional word2vec baselines. Finally, since all datasets comprise of a pair of words without any more con-
text, we are not comparing with any context-based represen-
tations.

Each of those embeddings is used as the Basis embed-
ding space in the Definition Encoder model, as described in
section 3.4. In our first experiments, we compare two ver-
sions of Definition Frames to the original Basis embeddings
without any fine-tuning or modification: one that contains all
the relations \( (DF_{\text{all}}) \) and one that contains only the word
and the \textit{isA} relation \( (DF_{\text{basic}}) \). Our choice of those Defini-
tion Frames is based on a series of ablation studies where we
eliminate dimensions. According to those studies, the \textit{isA}
relation affects the performance in a different way according
to the type of task (similarity versus relatedness).

In Tables 3 and 4 we summarize the results from those
experiments. Although we cannot clearly conclude whether
Definition Frames achieve better performance than the Basis
embeddings, we observe some interesting patterns of consist-
tent comparative performance (\textit{Basis}, \( DF_{\text{all}} \) and \( DF_{\text{basic}} \)).

Our first observation is that the comparative perfor-
mance of \( DF_{\text{basic}} \) and \( DF_{\text{all}} \) is mostly similar across all three Basis
embeddings for any given dataset. We further notice that for
many instances where \( DF_{\text{basic}} \) outperforms \( DF_{\text{all}} \), it also
outperforms \textit{Basis}. This consistent behavior indicates that,
although Definition Tensors carry additional useful informa-
tion through their structure, we do not exploit it in the best
way possible.

Our second observation concerns the difference on per-
formance with respect to the type of similarity. When we
compare the Definition Frames with the Basis embeddings
we notice that the former perform better in similarity tasks
(Table 3) than in relatedness (Table 4), as also reported by
Bosc and Vincent. Our explanation of the poor performance
in relatedness tasks is that, even if we have complete and
accurate information of all the relations, some relations are
not mapped properly due to the cosine similarity metric, a
problem also discussed by Faruqui et al. In our framework
for example, consider two highly related words like
\textit{car} and \textit{wheel}. Although Definition Frames might include the \textit{partOf}
relation between them, the standard cosine similarity metric
is not able to account for similarities across different dimen-
sions (in this case partOf with the actual word).

#### Applying a Linear Transform

In order to validate our hy-
pothesis about the effect of structure and whether the cosine
Table 3: Spearman’s correlation for word embeddings in similarity datasets. The best performing model between $DF_{all}$ and $DF_{basic}$ is shown in bold, while the best performing model overal ($Basis$, $DF_{all}$ and $DF_{basic}$) is underlined.

| Dataset  | GloVe      | Word2Vec wn | Dict2Vec |
|----------|------------|-------------|----------|
|          | Basis      | $DF_{all}$  | $DF_{basic}$ | Basis      | $DF_{all}$  | $DF_{basic}$ | Basis      | $DF_{all}$  | $DF_{basic}$ |
| RG-65    | 0.79       | 0.68        | 0.81      | 0.40       | 0.06        | 0.23        | 0.84       | 0.73        | 0.86        |
| SimLex   | 0.31       | 0.30        | 0.31      | 0.13       | 0.12        | 0.12        | 0.41       | 0.36        | 0.39        |
| SimVerb  | 0.19       | 0.14        | 0.15      | 0.13       | -0.02       | 0.04        | 0.24       | 0.14        | 0.16        |
| WS-SIM   | 0.63       | 0.57        | 0.65      | 0.58       | 0.47        | 0.52        | 0.78       | 0.76        | 0.78        |
| MC30     | 0.70       | 0.72        | 0.71      | 0.25       | 0.38        | 0.13        | 0.75       | 0.80        | 0.74        |

Table 4: Spearman’s correlation for word embeddings in relatedness datasets. The best performing model between $DF_{all}$ and $DF_{basic}$ is shown in bold, while the best performing model overal ($Basis$, $DF_{all}$ and $DF_{basic}$) is underlined.

| Dataset  | GloVe      | Word2Vec wn | Dict2Vec |
|----------|------------|-------------|----------|
|          | Basis      | $DF_{all}$  | $DF_{basic}$ | Basis      | $DF_{all}$  | $DF_{basic}$ | Basis      | $DF_{all}$  | $DF_{basic}$ |
| MEN      | 0.72       | 0.66        | 0.71      | 0.47       | 0.35        | 0.43        | 0.74       | 0.70        | 0.73        |
| MTurk287 | 0.74       | 0.68        | 0.70      | 0.30       | 0.25        | 0.28        | 0.71       | 0.66        | 0.72        |
| MTurk771 | 0.64       | 0.64        | 0.63      | 0.37       | 0.40        | 0.38        | 0.71       | 0.69        | 0.69        |
| RW-STAN  | 0.37       | 0.36        | 0.36      | 0.21       | 0.16        | 0.14        | 0.41       | 0.40        | 0.41        |
| WS-REL   | 0.51       | 0.46        | 0.52      | 0.35       | 0.31        | 0.36        | 0.67       | 0.62        | 0.67        |

similarity metric is an impediment for our representations, we design a slightly modified version of the previous experiments. For any dataset, instead of directly evaluating the encoded Definition Frame, we first apply a linear transformation on it. Thus, given the Definition Frames $DF_1$ and $DF_2$ for a pair of words $w_1$, $w_2$, we get

$$DF_1^* = W \times DF_1 + b$$

$$DF_2^* = W \times DF_2 + b$$

which we now use in our experiments. The parameters $W, b$ are learnt for each dataset separately to account for discrepancies across datasets. Our objective is to minimize the mean squared error between the cosine similarity of the linearly transformed representations and the normalized ground truth similarity score.

For our experiments we use 10-Fold cross-validation and we report the average performance. We ignore datasets with less than 100 instances due to their small size. We also follow the same method for the Basis embeddings on each dataset by learning the parameters $W_{basis}, b_{basis}$. In Table 5 we compare the performance of the Basis embeddings before and after the linear transformation ($Basis$ and $Basis^*$), with the Definition Frame ($DF$ and $DF^*$). Since they were the best performing embeddings in the previous section, we perform experiments with both GloVe and dict2vec as the Basis embeddings used for $Basis$ and $DF$. The performance of the embeddings before and after the transformation is reported on the same cross-validation splits to avoid randomness. Finally, for our reported results, we ignore datasets where both $DF$ and $Basis$ embeddings show lower performance after the linear transformation (MTurk287, MTurk771 and RW-STAN) or with a high $p$-value ($p > 0.05$) for the cross validation splits (SimVerb), as this hints inconsistency of the type of similarity within the dataset.

Our results show that $DF^*$ outperforms $Basis^*$ in most datasets. Furthermore, the average gain in performance (Gain) is significantly higher for Definition Frames, which confirms our previous hypothesis. We also report the performance after training jointly on all similarity (Sim-All) and relatedness datasets (Rel-All). In this setting we also include the small sized datasets (MC-30 and RG-65), but not those that show negative gain. Since we now have more data, we see a clear improvement of Definition Frames for both GloVe and dict2vec used as $Basis$. We further observe that for large datasets (Sim-All, Rel-All and MEN) $p$-values are extremely small ($p < 10^{-7}$) and $DF$ clearly outperforms $Basis$, whereas for smaller datasets (WS-SIM and WS-REL) $p$-values are higher ($p = 0.01$).

Through these experiments we show that structure leads to more fluid representations: a crucial factor when we need only a subset of the information encoded. Although fine-tuning is a widely used method to account for such phenomena, it typically involves complex models that require a lot of in-domain data. However, using only a linear transform, we achieve overall better performance compared to state-of-the-art pre-trained embeddings. This is a crucial step, as a linear transform allows to maintain semantic coherence of the representations compared to currently non-trackable neural methods.

4.2 Semantics of Definition Frames

The major contribution of Definition Frames is that, besides having overall better performance than other distributed representations, they are also semantically meaningful. While in the previous section we presented our results on their performance on word-similarity tasks, here we focus on their semantic aspect.

The first point to discuss is the quality of Definition Frames as a concept representation. Definition Frames are based on a set of relations and related terms that are ex-
Table 5: Spearman's correlation for embeddings before and after the linear transform. We show in bold the model with the best performance and the highest Gain. We denote with * the experiments with p-value $p < 0.001$.

Table 6: Amazon Turkers evaluation of Definition Frames.

In order to compare the output of the two systems, we asked from Amazon Mechanical Turkers to rank them (3 annotators per sentence). For each datum, we provide the original definition sentence, the Concept-query and the output of the two systems. Then, we ask each annotator the following question with three possible, mutually exclusive replies:

'Which system better represents the definition of the Concept-query?'
(1) system 1
(2) system 2
(3) both are equally good/bad.

In order to interpret the annotators' replies, we label a datum to belong to system $i$ if at least 2/3 of the annotators choose it, otherwise we label it to belong to the class equally good/bad. As we see in Table 6, according to the study, Definition Frames outperformed OpenIE by a large margin. Although we do not claim that our representation is better than OpenIE in a general setting (they have a different objective), these results are a good verification that Definition Frames are able to capture the semantics of definitions.

The second point is whether Definition Frames are still an explainable representation, after they are encoded in a matrix format. As discussed earlier, Definition Frames maintain a very specific structure. Given a concept $C$, each dimension of its Definition Frame contains the terms that are related with $C$ via a particular relation. The exact same structure is maintained in the matrix representation, as each row contains the now distributed representation of those same terms. Thus, from a human perspective, given a Definition Frame in a matrix format, we know that for every row $i$ that contains a non-zero vector, there is some term(s) that are related with $C$ via the relation $r_i$.

An important property of the Definition Frames is that we can retrieve those related terms from the matrix representation. As described in section 3.4, the Definition Encoder module maps each word to some pre-existing embedding space (Basis). Given that we do neither learn nor modify this space, we can easily find the word given its embedding or use any standard similarity metric (i.e. cosine distance, euclidean distance, etc) when multiple words are encoded in the same row. Thus, although the encoded matrix representation is not interpretable by humans as-is, we can easily convert it back to the original, semantically meaningful Definition Frame. This is also the reason why we only used a linear transformation in the second set of experiments in section 4.1, we can easily revert linear transformations, unlike the non-linearities of neural networks.

5 Conclusion & Future Work

Through this paper we propose a hybrid representation that has interpretable dimensions, while still maintaining properties of distributional semantics. While previous work focused on improving the performance of distributional vectors by infusing semantic knowledge in them, our goal is to design a novel representation that benefits from the information encoded in word embeddings but is also semantically meaningful. Towards this end, we achieve better results in word similarity tasks by using only a weighted version of our structured representations (linear transformation).

More than the representations themselves, the contribu-
tion of this work is that it sets a possible basis to combine meaning with downstream performance in NLP. Some promising directions for future work include improving the encoding of Definition Frames to a richer representation and exploring in depth how we can exploit the structure of Definition Frames to improve the representations. Another path of future work may focus on using the information encoded in Definition Frames to propagate information across them and to learn a new embedding space. Finally, we believe that Definition Frames can be an extremely useful representation to tasks that rely heavily in semantics, like common sense reasoning, open question answering, natural language inference, etc. Due to the nature of those tasks and their complexity, a hybrid meaningful distributed representation, like Definition Frames, allows us to choose which aspects of the representation are important for a problem.

References

[Agirre et al. 2009] Agirre, E.; Alfonseca, E.; Hall, K.; Kravalova, J.; Paśca, M.; and Soroa, A. 2009. A study on similarity and relatedness using distributional and wordnet-based approaches. In Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, 19–27. Association for Computational Linguistics.

[Anke and Schockaert 2018] Anke, L. E., and Schockaert, S. 2018. Syntactically aware neural architectures for definition extraction. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), 378–385.

[Baker, Fillmore, and Lowe 1998] Baker, C. F.; Fillmore, C. J.; and Lowe, J. B. 1998. The berkeley frame net project. In Proceedings of the 17th International Conference on Computational Linguistics-Volume 1, 86–90. Association for Computational Linguistics.

[Binot and Jensen 1993] Binot, J.-L., and Jensen, K. 1993. A semantic expert using an online standard dictionary. In Natural Language Processing: The PLNLP Approach. Springer. 135–147.

[Bojanowski et al. 2017] Bojanowski, P.; Grave, E.; Joulin, A.; and Mikolov, T. 2017. Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics 5:135–146.

[Bosc and Vincent 2018] Bosc, T., and Vincent, P. 2018. Auto-encoding dictionary definitions into consistent word embeddings. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 1522–1532.

[Bruni et al. 2012] Bruni, E.; Boleda, G.; Baroni, M.; and Tran, N.-K. 2012. Distributional semantics in technicolor. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1, 136–145. Association for Computational Linguistics.

[Calzolari 1984] Calzolari, N. 1984. Detecting patterns in a lexical data base. In 10th International Conference on Computational Linguistics and 22nd Annual Meeting of the Association for Computational Linguistics.

[Chodorow, Byrd, and Heidorn 1985] Chodorow, M. S.; Byrd, R. J.; and Heidorn, G. E. 1985. Extracting semantic hierarchies from a large on-line dictionary. In Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics, 299–304. Association for Computational Linguistics.

[Devlin et al. 2018] Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[Faruqui and Dyer 2014] Faruqui, M., and Dyer, C. 2014. Community evaluation and exchange of word vectors at wordvectors.org. In Proceedings of ACL: System Demonstrations.

[Faruqui et al. 2015] Faruqui, M.; Dodge, J.; Jauhar, S. K.; Dyer, C.; Hovy, E.; and Smith, N. A. 2015. Retrofitting word vectors to semantic lexicons. In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 1606–1615.

[Faruqui et al. 2016] Faruqui, M.; Tsvetkov, Y.; Rastogi, P.; and Dyer, C. 2016. Problems with evaluation of word embeddings using word similarity tasks. In Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP, 30–35.

[Finkelstein et al. 2002] Finkelstein, L.; Gabrilovich, E.; Matias, Y.; Rivlin, E.; Solan, Z.; Wolfman, G.; and Ruppin, E. 2002. Placing search in context: The concept revisited. ACM Transactions on Information Systems 20(1):116–131.

[Gábor et al. 2018] Gábor, K.; Buscaldi, D.; Schumann, A.-K.; QasemiZadeh, B.; Zargayouna, H.; and Charnois, T. 2018. SemEval-2018 task 7: Semantic relation extraction and classification in scientific papers. In Proceedings of The 12th International Workshop on Semantic Evaluation, 679–688.

[Gerz et al. 2016] Gerz, D.; Vulić, I.; Hill, F.; Reichart, R.; and Korhonen, A. 2016. Simverb-3500: A large-scale evaluation set of verb similarity. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 2173–2182.

[Halawi et al. 2012] Halawi, G.; Dror, G.; Gabrilovich, E.; and Koren, Y. 2012. Large-scale learning of word relatedness with constraints. In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, 1406–1414. ACM.

[Hendrickx et al. 2009] Hendrickx, I.; Kim, S. N.; Kozareva, Z.; Nakov, P.; Ó Séaghdha, D.; Padó, S.; Pennacchiotti, M.; Romano, L.; and Szpakowicz, S. 2009. SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions, 94–99. Association for Computational Linguistics.

[hill et al. 2016] Hill, F.; Cho, K.; Korhonen, A.; and Bengio, Y. 2016. Learning to understand phrases by embedding
the dictionary. Transactions of the Association for Computational Linguistics 4:17–30.

[Hill, Reichart, and Korhonen 2015] Hill, F.; Reichart, R.; and Korhonen, A. 2015. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. Computational Linguistics 41(4):665–695.

[Hochreiter and Schmidhuber 1997] Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. Neural computation 9(8):1735–1780.

[Lenat 1995] Lenat, D. B. 1995. Cyc: A large-scale investment in knowledge infrastructure. Commun. ACM 38(11):33–38.

[Luong, Socher, and Manning 2013] Luong, T.; Socher, R.; and Manning, C. 2013. Better word representations with recursive neural networks for morphology. In Proceedings of the Seventeenth Conference on Computational Natural Language Learning, 104–113.

[Ma and Hovy 2016] Ma, X., and Hovy, E. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), volume 1, 1064–1074.

[Manning et al. 2014] Manning, C.; Surdeanu, M.; Bauer, J.; Finkel, J.; Bethard, S.; and McClosky, D. 2014. The stanford corenlp natural language processing toolkit. In Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations, 55–60.

[Mikolov et al. 2013] Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems, 3111–3119.

[Miller and Charles 1991] Miller, G. A., and Charles, W. G. 1991. Contextual correlates of semantic similarity. Language and cognitive processes 6(1):1–28.

[Miller 1995] Miller, G. A. 1995. Wordnet: a lexical database for english. Communications of the ACM 38(11):39–41.

[Mrkšić et al. 2016] Mrkšić, N.; OSéaghdha, D.; Thomson, B.; Gašić, M.; Rojas-Barahona, L.; Su, P.-H.; Vandyke, D.; Wen, T.-H.; and Young, S. 2016. Counter-fitting word vectors to linguistic constraints. In Proceedings of NAACL-HLT, 142–148.

[Mrkšić et al. 2017] Mrkšić, N.; Vulić, I.; Séaghdha, D. Ó.; Leviant, I.; Reichart, R.; Gašić, M.; Korhonen, A.; and Young, S. 2017. Semantic specialization of distributional word vector spaces using monolingual and cross-lingual constraints. Transactions of the Association for Computational Linguistics 5:309–324.

[Noraset et al. 2017] Noraset, T.; Liang, C.; Birnbaum, L.; and Downey, D. 2017. Definition modeling: Learning to define word embeddings in natural language. In Thirty-First AAAI Conference on Artificial Intelligence.

[Pennington, Socher, and Manning 2014] Pennington, J.; Socher, R.; and Manning, C. 2014. Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 1532–1543.

[Peters et al. 2018] Peters, M.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018. Deep contextualized word representations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), 2227–2237.

[Radinsky et al. 2011] Radinsky, K.; Agichtein, E.; Gabrilovich, E.; and Markovitch, S. 2011. A word at a time: computing word relatedness using temporal semantic analysis. In Proceedings of the 20th international conference on World wide web, 337–346. ACM.

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

[Speer and Havasi 2012] Speer, R., and Havasi, C. 2012. Representing general relational knowledge in conceptnet 5. In LREC, 3679–3686.

[Stanovsky et al. 2018] Stanovsky, G.; Michael, J.; Zettlemoyer, L. S.; and Dagan, I. 2018. Supervised open information extraction. In NAACL-HLT.

[Tissier, Gravier, and Habrard 2017] Tissier, J.; Gravier, C.; and Habrard, A. 2017. Dict2vec: Learning word embeddings using lexical dictionaries. In Conference on Empirical Methods in Natural Language Processing (EMNLP 2017), 254–263.

[Vulić and Mrkšić 2018] Vulić, I., and Mrkšić, N. 2018. Specialising word vectors for lexical entailment. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), 1134–1145.

[Zock and Bilac 2004] Zock, M., and Bilac, S. 2004. Word lookup on the basis of associations: From an idea to a roadmap. In Proceedings of the Workshop on Enhancing and Using Electronic Dictionaries, ElectricDict ’04, 29–35. Stroudsburg, PA, USA: Association for Computational Linguistics.