Ágoston Tóth:
Recognizing semantic frames using neural networks and distributional word representations
Argumentum 14 (2018), 400-414
Debreceni Egyetemi Kiadó

Tanulmány

Ágoston Tóth

Recognizing semantic frames using neural networks and distributional word representations

Abstract

This paper reports the results of a series of experiments into recognizing semantic frames and frame elements using neural networks and measuring the added benefit of embedding large-scale co-occurrence information about words during the process. Frame recognition is carried out using Elman-type recurrent neural networks to give the system short-term memory of previous words within the sentence. Long-term memory is implemented in the system of weighted links between neurons. We test 9 word-representation methods including predict- and count-type distributional representations. We show that distributional word representations, which provide the frame recognizer with access to unlabelled co-occurrence information about every word, perform noticeably better than non-distributional techniques. Frame recognition F-score increased from 0.76 to 0.89, and frame element recognition – a considerably more difficult task – also benefited from the added information: we see an F-score increase from 0.46 to 0.53. We also show that this task is less sensitive to the particularities of collecting word distribution information than the known benchmark experiments.
Keywords: FrameNet, semantic role labelling, distributional semantics, word embeddings, deep learning

1 Introduction

This paper documents a series of experiments in which custom-made artificial neural networks assign frame-semantic labels taken from FrameNet (Baker, Fillmore & Lowe 1998) to words in sentences, and we also investigate whether this task benefits from embedding co-occurrence information about words captured by different types of distributional word representation methods trained on large unannotated corpora.

The structure of the paper is as follows. *Section 2* provides the reader with relevant information about the following areas: frame semantics in FrameNet (*2.1*), the basics of using neural networks for language processing (*2.2*), creating distributional word representations that encode co-occurrence information about words (*2.3*) and producing distributional word representations inside neural networks that predict the context of a word or the word based on its context (*2.4*). *Section 3* describes my experiments by explaining the methodological steps first (*section 3.1.1*: details about the semantic frame recognition task, *3.1.2*: detailed information about the selected word representation methods, *3.1.3*: brief description of the necessary software infrastructure), then *section 3.2* reports the results. *Section 4* contains my concluding remarks.

Ágoston Tóth:
Recognizing semantic frames using neural networks and distributional word representations
Argumentum 14 (2018), 400-414
Debreceni Egyetemi Kiadó

2 The background

2.1 Frame semantics in FrameNet

FrameNet (FN; Baker, Fillmore & Lowe 1998) is a large-scale semantic database that relies on the notion of semantic frame, which is like a script that characterizes the type of the situation or event (by categorizing it into a frame) and identifies the participants and "props" of the semantic frame (these are the frame elements, FEs). Through investigating corpus evidence, the editors locate lexical units that instantiate a frame and look for visible semantic arguments. Only those phrases are annotated that are related to the target word, but a single sentence may contain multiple frames, which results in multiple layers of annotation.

We get the following pieces of information from the frame specification for the *Leadership* frame, for example[1]:

- Definition: "These are words referring to control by a Leader over a particular entity or group (the Governed) or an Activity. The frame contains both nouns referring to a title or position (e.g. director, king, president), and verbs describing the action of leadership (e.g. rule, reign)…".
- Example sentences with FEs highlighted.
- Core frame elements with definitions and/or examples: *Activity*, *Governed*, *Leader*, *Role*.
- Non-core frame elements: *Degree*, *Descriptor*, *Duration*, *Place*, *Time*, etc. (with definitions and/or examples).
- Relations between frames.
- Lexical units that evoke this frame, with part of speech information: *administration.n, authority.n, baron.n, bishop.n, boss.n, captain.n, CEO.n, chair.v, chairman.n, chairperson.n, charge.n, chief executive officer.n, chief.n, command.n, command.v, commander.n, dictator.n, rule.n, rule.v,* etc.

Notice that words that do not take syntactic arguments may also quality as frame-evoking lexical units.

The experiments discussed in this paper are based on FrameNet version 1.7. I have only used the "FrameNet continuous text annotation" part of the FrameNet database, which contains short documents with FN frame and frame element labels added to words.

2.2 Neural networks for language processing

At the heart of this frame recognition apparatus lies a connectionist machine learning device that uses Elman-type *recurrent neural networks* (Elman 1990) for solving the semantic task. The learning phase is *supervised* via the use of FrameNet frame and frame element annotations.

Neural networks are widely used in many areas of life (e.g. face recognition, medical diagnostics and diagnosis, stock price prediction, machine translation). They tolerate noise well and they are known to make useful generalizations using large datasets to make predictions for never-seen input patterns (due to noise or novel information – both of these factors are present in human communication). Artificial Neural Network models simulate the parallel processing nature of the human brain. The adaptive power of this highly inter-connected structure lies in the system of weighted connections between neurons and also in the training process: during training, a learning algorithm is used that gradually changes the

[1] taken from the FrameNet database (https://framenet.icsi.berkeley.edu/) on January 10, 2016

Ágoston Tóth:
Recognizing semantic frames using neural networks and distributional word representations
Argumentum 14 (2018), 400-414
Debreceni Egyetemi Kiadó

connection weights until the desired output is reached closely enough, for every training example. The desired output, in our case, will be the expected activation pattern of output neurons to indicate frame and frame element status as discussed in section 3.1.1.

The neural networks used in my experiments are Elman-type simple recurrent networks (Elman 1990). Here, they contain as many input neurons as needed for representing a single word on the input, a 25-unit hidden layer[2] and 7–13 neurons in the output layer for the frame and frame element prediction information. The networks also feature a small, 25-unit context layer that interacts with the hidden layer. The *hidden layer - context layer* Elman loop gives the network short-term memory[3]: output is produced based on the current input word as well as all the words that precede the current input in the sentence. As regards the output, one neuron is responsible for labelling the current input as a frame-evoking lexical unit and additional neurons react to the FE status of the word (one output neuron per frame element type).

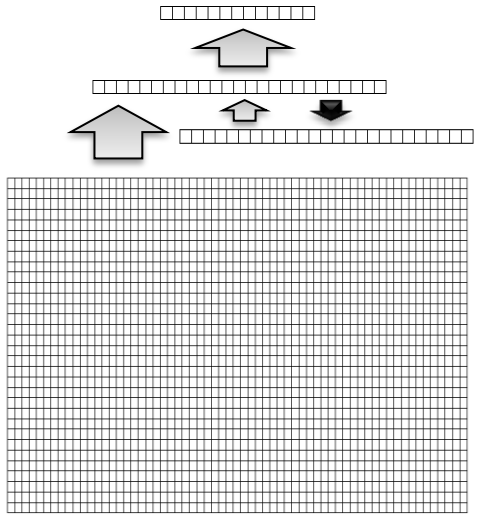

Figure 1: Neural network design. From bottom to top: input group (representation of current input word), context and hidden groups (25 unit transformation + 25 unit temporal memory), output group (frame and frame element information).

Representing words in the input layer of a neural network has always been a challenge. Some early systems worked with fixed-length words only, which was, of course, a major limitation. Consider McClelland and Rumelhart (1981) and Bullinaria (1995) as examples: the former system worked with four-letters words exclusively, while the latter was restricted to taking monosyllabic words of one onset consonant cluster, one vowel cluster and one offset consonant cluster. The traditional one-hot representation is similarly straightforward, but limited in use: a single neuron, which stands for a word (lemma or word form), is activated, the rest of the input layer remains inactive. In this paper, I have compared 9 different techniques for representing words on the input of the neural network (see section 3.1.2), including those that supply the network with information about the distributional properties of the current word.

[2] A *layer* is a group of neurons that have a similar function in the network.
[3] Whereas *long-term* memory is implemented in the system of weighted links between neurons.

Ágoston Tóth:
Recognizing semantic frames using neural networks and distributional word representations
Argumentum 14 (2018), 400-414
Debreceni Egyetemi Kiadó

2.3 Capturing the distributional properties of words through counting selected words in its context

Many of the word representations I test employ *distributional feature vectors* computed from a large corpus in a distinct early step of processing in the following way. Each *target word* (every word that appears in the frame-semantic task) is represented in a multi-dimensional space using a vector. Each component of this vector signals or counts the number of co-occurrences of the given target word with one of the context words we use for characterizing target items. For example, if the word *drink* is a target word and the word *tea* is among the context words that we keep track of, and *tea* occurs 23 times in the close vicinity (in the *context window*) of *drink*, then the vector component corresponding to the word *tea* (in the feature vector describing the word *drink*) will be set to 23.

$v_{drink} = < freq_1, freq_2, \ldots, 23, \ldots, freq_t >$ where t=total number of context words

Large corpora are necessary for building useful distributional feature vectors (cf. Bullinaria & Levy 2007). "Raw", unprocessed corpora are suitable for the task, but annotation can also be taken into account (e.g. part of speech categories). Optionally, the components of the vector can be weighted so that unusual or "surprising" co-occurrence events become more salient. Logarithmic weighting of the components is fast and useful because it will prevent very frequent context words from supressing the effect of less frequent context words. Another way of weighting the vector components is replacing positive Pointwise Mutual Information (pPMI) scores for the original frequency values. This process emphasizes less probable co-occurrence events over more probable ones thereby reflecting the *significance* of a co-occurrence rather than its raw frequency.

With the feature vectors in hand, we can compare the target words. Although the experiments in this paper do not directly compare distributional feature vectors, vector comparison (via measuring vector distance or calculating their cosine) is a great tool for exploiting distributional data and it makes semantic, morphological and other relations between words observable. Unfortunately, all these relations appear in the same space with no obvious way to tell them apart. To exemplify the resulting situation, table 1 shows the 19 distributionally most similar words to the Hungarian word *kis* ('little', 'tiny').

Rank	*Similar word*	*Typical English equivalents*	*Similarity score*
1	nagy	big, large	0.413
2	kisebb	smaller	0.376
3	nagyobb	bigger, larger	0.347
4	hatalmas	huge, enormous, vast	0.32
5	apró	tiny, minuscule	0.3
6	sok	many, much	0.296
7	egy	a, an, one	0.291
8	a	the	0.282
9	kicsi	tiny, small, little	0.265
10	olyan	such, so	0.264
11	legnagyobb	biggest, largest	0.258
12	szép	nice, pretty, beautiful	0.257
13	ilyen	such a(n), so	0.253
14	másik	other	0.25
15	kevés	little	0.242
16	két	two	0.241
17	egész	all, whole, complete	0.237
18	óriási	gigantic, giant, enormous	0.237
19	legtöbb	most	0.223

Table 1: Words most similar to kis ('little', 'tiny') (from Tóth 2014:40)[4]

The distribution of the adjective *nagy* ('big', 'large') has been found most similar to the distribution of *kis* ('little', 'tiny'). The list includes other antonyms, too (*hatalmas, sok, óriási*). Synonyms are also present (*kicsi, kevés*), as well as the comparative form of *kis* (*kisebb*), which is the second most similar item to the initial target word. The superlative form, *legkisebb*, is the 26th item on the list (therefore, it is not shown above), but its score is still relatively high. It is a general observation that words that can be related to the target word through the established lexical semantic relations (synonymy, antonymy, hyponymy/hypernymy) do appear among the results, but we cannot easily distinguish among these relations using feature vector comparisons only.[5]

Distributional representations have been shown to have psycholinguistic relevance, too. Among others, Pado & Lapata (2007) use distributional feature vectors to characterize prime and target words featured in Hodgson's (1991) dataset (143 prime-target pairs), then they compare the vectors in each pair. The idea is that those word pairs that exhibit priming behaviour in Hodgson (1991) should appear more distributionally similar than the average of generated pairs containing unrelated prime – target word pairs. Pado and Lapata show that

[4] Similarity values and the similarity rank are sensitive to parameters such as context vocabulary size, context window size, vector component weighting and comparison method.

[5] One of the few attempts at doing so is Scheible, Schulte im Walde and Springorum (2013); they try to distinguish between synonymy and antonymy using distributional data only.

Ágoston Tóth:
Recognizing semantic frames using neural networks and distributional word representations
Argumentum 14 (2018), 400-414
Debreceni Egyetemi Kiadó

related primes are significantly more similar to their targets than unrelated primes are in all six subclasses of the dataset.

See Tóth (2014) for a more detailed overview of distributional semantics.

2.4 Capturing the distributional properties of words through predicting context

Mikolov et al. (2013a) described an indirect method of gathering distributional information about words in a machine learning environment. Instead of counting co-occurrence information for target and context words, their neural network learns to predict word context using either of the following two approaches:

a) The network learns to predict the most probable target word from a *n*-word context window. The input contains a simple numerical representation of exactly *n* words, the output should contain the numerical representation of the target word. The authors call this model the continuous bag-of-words (*CBOW*) model.
b) The network learns to predict the *n*-word context on its output when it receives a numerical representation of a single know word on its input; this is called the *Skip-gram* model.

The network topology is simple: the input layer contains the units that encode the context words (for CBOW) or the target word (Skip-gram); activation information then propagates to the hidden layer of neurons (50–600 units in most experiments) through trainable weighted links; then activation goes to the output layer that encodes the target word (CBOW) or the context words (Skip-gram) through another set of trainable weighted links. The network learns to minimize error, but the actual final accuracy of the prediction task is not a major concern; the point of the whole process is to develop internal word representations ("word embeddings") in the hidden layer of the network during the process. The similarity of these internal representations is considered to result from the similarity of the contextual distribution of target words. Mikolov et al. (2013a) showed that these word representations can be exploited for solving linguistically meaningful tasks, such as getting the past tense of a verb (*walking - walked + swimming = X*, where *X* should be most similar to the representation of *swam*), plural (*dollar – dollars + mouse ~ mice*) or semantic analogy tasks (*Athens - Greece + Oslo ~ Norway*). If embeddings are treated as vectors, then the above calculations can be carried out using vector operations and similarity can be calculated as the cosine of the angle between vectors. The authors reported a maximum of 55% accuracy for their 8869 semantic questions and 64% accuracy for 10675 syntactic questions. They used a large, 1-million-word target word vocabulary and a 6-billion-word training corpus.

2.5 Related literature

Non-connectionist works that utilize distributional data in semantic analysis include Pennacchiotti et al. (2008) and Hermann et al. (2014). Pennacchiotti et al. (2008) employ distributional word representations for lexical unit induction, i.e. for extending FrameNet's scope by covering more frame-evoking lexical units. The underlying technique is word-similarity measurement through the comparison of distributional features. Hermann et al. (2014) also rely on distributional semantics in their proposal of a two-stage process of frame-semantic parsing. In the first stage, frame identification and disambiguation are carried out. In the second stage, frame elements are identified.

Ágoston Tóth:
Recognizing semantic frames using neural networks and distributional word representations
Argumentum 14 (2018), 400-414
Debreceni Egyetemi Kiadó

As far as word representations are concerned, Baroni, Dinu & Kruszevski (2014) gave a thorough and systematic comparison of several distributional word representation methods. They compared them using a variety of benchmarks, including semantic relatedness tasks, synonym detection, concept categorization, selectional preferences tests and analogy tasks. Before their study, the traditional distributional representations (section 2.3) and the novel context-predicting representations (section 2.4) had not been systematically compared under the same conditions. They were surprised to find that the "buzz" around the context-predicting models was justified: these models were superior to the counting models across various tests.

3 Experiments

3.1 Methods

3.1.1 Semantic frame and frame element labelling

I carried out a series of frame and frame element labelling experiments with the FrameNet frames shown in Table 2.

FrameID	*Frame name*	*Frequency rank*	*Frame instances*	*Number of FE types*
73	Leadership	5	499	13
173	Buildings	7	420	12
408	Manufacturing	14	277	13
191	Natural_features	16	269	9
118	Possession	17	260	7
990	Capability	18	259	8
304	People	19	257	8
34	Discussion	81	87	12
1371	Organization	89	79	8
141	Certainty	93	74	7
172	Commerce_sell	95	73	8
171	Commerce_buy	145	50	9

Table 2: FrameNet frames and frame elements in the experiments[6]

To put these values in context: there are 792 different frame types and a total of 28783 frame instances in the FrameNet full-text corpus. The experiments cover about 9% of the frame instances, with a selection of frames that contains some very frequent and some middle-frequency semantic frames.

Some of the frame elements that the system learns to predict are the following (for illustration only):

- Leadership frame: *Leader*, *Role*, *Governed*, etc.
- Buildings frame: *Name*, *Type*, *Possessor*, etc.

[6] as found in FrameNet's full-text corpus, version 1.7 (https://framenet.icsi.berkeley.edu/)

- Possession frame: *Owner*, *Possession*, etc.
- People frame: *Origin*, *Ethnicity*, *Age*, etc.
- Commerce_buy frame: *Goods*, *Buyer*, *Seller*. etc.

One half of the relevant sentences was assigned to the training set, the other half to the testing set. To cross-validate the experiment, the testing and training corpora were swapped and re-evaluated, and the results were averaged using arithmetic mean.

Average sentence length was 21 words. Sentences that contained more than 40 words (very rare) or less than 2 words were excluded from processing. Tokenization was not performed, multi-word units were not grouped together – in general, no linguistic processing was applied to the training and testing data. Only the frame and frame element labels were retrieved from the FrameNet corpus and used as training or testing targets.

I trained and tested 12 neural networks. Each neural network was responsible for recognizing a single semantic frame by pinpointing the frame-evoking words (Lexical Units, LUs) and the frame elements in the sentence. LU status was indicated by a single output neuron. Frame elements were recognized by setting a frame-recognition flag (a designated neuron) in the appropriate frame-specific network in addition to activating the output neuron corresponding to the frame element. A positive recognition event (true positive or false positive) was recorded when either the LU unit or the frame master switch + a FE unit combination reached a designated threshold (50% activation level). When two or more FE output neurons reached the threshold at the same time (for the same input word), the output was treated as a false positive, which decreased the precision. When the frame-recognizing master switch remained inactive, it was interpreted as a valid recognition event (with negative outcome: "no label"), which could be a true negative or a false negative.

To measure and improve the accuracy of the network during *training*, outputs are evaluated constantly using an error measure, which is a function of the expected output and the actual response of the network. Training (i.e. the adjustment of connection weights between neurons) is carried out using the 'backpropagation through time' (SRBPTT) algorithm, which relies on a single backpropagation process at the end of each sentence rather than an individual sweep after each word. In this way, the network considers *all words* of the current sentence during training (for changing the connection weights) to get the best possible output for each word of the sentence. Each network was trained in 1200 passes, with one sentence taken from the training corpus in random order in each pass. The network saw every sentence at least twice; this number depended on the number of available training examples for the given frame (see table 2).

During training, precision and recall were high, with F-scores in the 0.9–1 interval, which means that the training dataset was learnable and the selected network architecture worked well. The high training accuracy also means that the semantic task was not severely affected by lexical ambiguity, probably due to the frame-specific nature of the individual semantic processor networks. The networks were tiny in size (25-unit hidden layers) to make them generalize data and not simply learn the examples one-by-one.

Precision and recall for unseen *testing* sentences were computed to assess the ability of the networks to label new sentences with new patterns and words not seen during frame-labelling training. The results will be shown and discussed in section 3.2.

Feed-forward and recurrent networks with different hidden layer dimensions (10, 50, 100, 300 neurons) have also been tested – with less success. I only report data for the network setup that I have found optimal.

Ágoston Tóth:
Recognizing semantic frames using neural networks and distributional word representations
Argumentum 14 (2018), 400-414
Debreceni Egyetemi Kiadó

3.1.2 Word representation methods

I created 9 different representations for each word type of the full-text FrameNet corpus (the *target words* of the investigation) in a separate step of the experiment. When the distributional properties of target words had to be computed, a 100-million-word subcorpus of the TC Wikipedia corpus[7] was used for getting information about the contexts in which these words typically occurred. I also added the sentences of the FrameNet full-text corpus (with no linguistic annotation in this case) to make sure that we got some contextual information for each target word, even for hapax legomena. The tested representation methods were as follows:

a. COUNT-LOG: Each word type of the FrameNet full-text corpus was represented by a distributional feature vector. Each representation was a vector with 5000 components (one component per context word) in which each component was normalized to the [0,1] interval using a logarithmic function[8].
b. COUNT-PPMI: Similar to COUNT-LOG, but vector components were weighted using positive Pointwise Mutual Information (pPMI): $I+(c;t)=\max(0,\log(P(c|t)/P(c)))$, where *c* is a context word, *t* is the target word, P(c|t) is the conditional probability of *c* given *t* and P(c) is the probability of *c*. pPMI emphasizes less probable co-occurrence events over more probable ones thereby reflecting the significance of a co-occurrence rather than its raw frequency. Components were normalized to [0,1].
c. RND-PPMI: COUNT-PPMI feature vectors shuffled in the following way: two words were picked using a random number generator, their representations were swapped and the process was repeated for several thousand word pairs. These representations were not *distributional* any more, since they did not hold real information about the context of the given word. They remained *distributed* representations, however, since information was distributed over several input units (compare it to the 1HOT representation, which is non-distributed). The RND-PPMI representation gives us a baseline for evaluating the COUNT-PPMI representation.
d. PRED-SKIPGRAM: Instead of counting co-occurrence information for target and context words, a neural network learned to predict the context of the words (see section 2.4) in a subcorpus of the TC Wikipedia corpus mentioned above. The representations were generated using the word2vec program (Mikolov et al. 2013b). For the Skip-gram representation, the algorithm learned to predict the most probable context having seen a single word (the target word) on the input. After training the Skip-gram network, hidden-layer activations for every target word were extracted from the system and saved as the PRED-SKIPGRAM representation for the given word. Each representation was a vector of 300 real numbers.
e. PRED-CBOW: It is similar to PRED-SKIPGRAM, but word2vec learns to predict the most probable target word (the expected output) for an *n*-word context window (the input). After the training phase, hidden-layer activations observed for target words are saved as word representations. Size: 300 components per representation.
f. RND-SKIPGRAM: PRED-SKIPGRAM embeddings shuffled using a random function: they resemble the original vectors but do not encode distributional

[7] http://nlp.cs.nyu.edu/wikipedia-data

[8] When components were normalized using a *linear* function, the most prominent co-occurrence events got a high activation level (near 1), but most events became difficult to represent (they got an activation of or close to 0). I tested this option, too, but the results were unremarkable.

information any more. This representation is a baseline for assessing the additional distributional information in the PRED-SKIPGRAM representation.

g. 1HOT: one-unit-per-word ("one-hot", localist) representation, which identifies each word type of the FrameNet full-text corpus using a unique input pattern in which exactly one neuron is activated (set to 1). The remaining input neurons remain inactive (0), i.e. the representation is non-distributed. This representation method does not encode information about the contexts in which words occur (non-distributional). It can be seen as a baseline for all the other representation methods as the one-hot representation is a common way of representing words in neural networks.
h. 1HOT+PPMI: the 1HOT representation of the target word plus its COUNT-PPMI representation, in one vector.
i. 1HOT+PRED-SKIPGRAM: a combined 1HOT and PRED-SKIPGRAM representation for each word.

Distributional feature vectors for the COUNT-LOG and COUNT-PPMI representations were collected from the above-mentioned subcorpus of the TC Wikipedia corpus. I used the 5000 most frequent words of TC Wikipedia as *context words* to characterize each word type (*target word*) that occured in the FrameNet continuous text annotation dataset. The inspected context was a 3+3 symmetrical rectangular window around the target words. Linguistic annotation was not used, frequent words were not filtered out.

The selected word representation method also had an impact on the frame-recognizer neural network. The general architecture of the network is described in section 2.2; figure 1 shows the outline. The size of the *input* group reflects *a)* the maximum vocabulary size in ID-representations (5000), *b)* the exact number of context words for characterizing target words in COUNT- representations (5000), or *c)* the size of the neural embeddings in the simulations based on PRED-* word embeddings (300 units). As far as combined representations are concerned, the 1HOT+PRED representation used 5300 input units, and the 1HOT+PPMI combination had the largest input patterns with 10000 input units. The size of the *output* group depended on the number of frame elements in the given semantic frame and it varied accordingly from network to network.

3.1.3 Tools of analysis

The *count-type representations* were generated by my own tool that processed Wikipedia data and generated the co-occurrence feature vectors. A ready-made program, *word2vec* (Mikolov et al. 2013b) was used to create the *predict-type representations*.

Frame and frame element recognition required software development, too. A custom-made program was needed to extract the training and testing data from the FrameNet full-text annotation corpus and produce the training and testing corpora readable by the neural-network simulator used in the next step. This program also substituted the original words by the numerical word representations (one-hot, distributional, etc.) producing 9 different training and 9 testing corpora (plus the alternative training/testing data for the randomized representations). All neural network simulations were performed using *LENS* (Rohde 1999) with custom-made scripts to set up the networks, to train and test them. Finally, I also wrote an evaluation script to compare the neural network output with the FrameNet frame and frame element labels present in the corpus to get precision and recall figures.

Ágoston Tóth:
Recognizing semantic frames using neural networks and distributional word representations
Argumentum 14 (2018), 400-414
Debreceni Egyetemi Kiadó

3.2 Results

Table 3 shows the precision, recall and F-score values measured in the semantic frame and frame element labelling experiments using different input representation methods. The table contains testing results rather than training data to show the prediction capability of the networks and not their capacity to memorize labels for known sentences.

Representation method	*p (FR)*	*r (FR)*	*F (FR)*	*p (FR+FE)*	*r (FR+FE)*	*F (FR+FE)*
1HOT	91	65	*76*	54	40	*46*
COUNT-LOGFREQ	91	86	***89***	56	43	*48*
COUNT-PPMI	93	85	*88*	56	47	*51*
RND-PPMI [9]	90	76	*82*	54	43	*48*
PRED-CBOW [10]	88	84	*86*	56	49	***53***
PRED-SKIPGRAM [11]	89	83	*86*	57	49	***53***
RND-SKIPGRAM [12]	81	68	*74*	50	40	*44*
1HOT+PPMI	92	85	*88*	58	46	*51*
1HOT+PRED-SKIPGRAM	90	86	*88*	56	49	***53***

Table 3: Effect of input representation method on precision (%), recall (%) and F-score (averages for the 12 semantic role recognizer networks; FR: frame labelling via Lexical Unit recognition; FR+FE: frame and frame element labelling)

3.2.1 1HOT vs. all the rest

The 1HOT representation was clearly inferior to all distributional representations (note that RND representations are not distributional, also see section 3.2.3). 1HOT is a general baseline, a common way of representing words in neural networks.

This representation method does not encode information about the context in which words occur. The network that carries out the semantic analysis processes words in context (it parses full sentences from the FrameNet corpus), which also means that contextual information is gathered during the process, but this context is limited to the few sentences FrameNet provides us with (see the 'Frame instances' column in table 2).

3.2.2 The added benefit of large-scale distributional information

One of the main questions addressed in this paper is whether (and to what extent) the addition of general, large-scale distributional information enhances the accuracy of frame and frame element recognition.

When we use 1HOT as the baseline, the best distributional techniques result in a considerable F-score gain: 13 percentage points in frame recognition and 7 points in frame element recognition.

We can also compare the following representations: RND-PPMI vs. COUNT-PPMI and RND-SKIPGRAM vs. PRED-SKIPGRAM. The RND versions are shuffled pPMI and Skip-

[9] average of 3 trial runs; baseline for COUNT-PPMI
[10] word2vec parameters: -cbow 1 -hs 0 -negative 5 -size 300 -window 3
[11] word2vec parameters: -cbow 0 -hs 0 -negative 5 -size 300 -window 3
[12] average of 3 trial runs; baseline for PRED-SKIPGRAM

gram representations, respectively, but by randomizing the feature vectors, we lose (all) distributional information. As regards the pPMI context vectors, there is a 6-point F-score difference in frame labelling and a 3-point difference in FE labelling between the randomized and the original distributional cases. For Skip-gram, we get a 12-point increase in frame labelling and a 9-point gain in frame element recognition over the randomized version (but RND-SKIPGRAM is much worse than RND-PPMI, also see section 3.2.3). Overall, the addition of distributional information collected from large corpora *does* effectively help frame recognition trained on a much smaller corpus. Since raw, unannotated corpora are relatively easy to compile, while semantically annotated corpora are expensive to produce, this insight is useful for future investigations and applications.

3.2.3 Identification only

In the lack of distributional information, the only function of RND representations is to separate the different words using randomized patterns, which makes them similar in function to the 1HOT representation. Of course, they are also quite different, since 1HOT patterns contain a single non-zero component with zeros in all other positions and are meant to produce 1:1 mappings between representations and word forms, while information in the RND representations is distributed over the entire group of input neurons and – for unrelated reasons – unique identification is not ensured. It was surprising to see the difference between these methods: 1HOT was noticeably better than RND-SKIPGRAM, but worse than RND-PPMI. A possible explanation is that neural processing may favour distributed representations with many active input neurons, but this distributed (but non-distributional) nature is not an advantage when the vectors are so similar to each other as the Skip-gram vectors produced by the word2vec tool.

3.2.4 Combined representations: identification + distributional data

Two of the tested representations contained a distributional component and also the one-hot representation for each word type for guaranteed unique identification. The COUNT-PPMI and the 1HOT+PPMI representations were nearly identical in performance in the frame recognition task, there was no improvement from the added 1HOT component. The 1HOT+PRED-SKIPGRAM representation was better than PRED-SKIPGRAM on its own, however: the additional one-hot component helped to make up for the deficiency mentioned in connection with the word2vec vectors in section 3.2.3. As far as frame element recognition was concerned, we did not see any performance gain from the added 1HOT component, however.

3.2.5 The discrepancy between existing benchmarks and the new results

Parameter tuning has its own literature in distributional semantics (e.g. Bullinaria & Levy 2007, Bullinaria & Levy 2012), and it is a general observation that the best parameter set changes from task to task. In the case of distributional feature vectors collected from corpora using the traditional "counting" technique, pPMI weighting and other statistical weighting methods tend to return superior results. Regarding the frame labelling task in this experiment, pPMI weighting did improve precision, but it also decreased recall when compared to simple

log(frequency) data. In the frame element labelling task, pPMI only slightly increased recall and did not affect precision.

A robust finding of Baroni, Dinu & Kruszevski (2014) is that predict embeddings work significantly better than count-type feature vectors in many tasks. It was not the case here in frame recognition: the best count method was better than the best predict method. As far as frame element labelling was concerned, however, predict methods were slightly better. When we consider predict methods only, we see that Skip-gram and CBOW embeddings performed very similarly.

I would also like to point out that these experiments seem to have been less sensitive to the particularities of representing word distribution than most benchmarks. I attribute this to the fact that, in our case, semantic analysis is carried out using neural networks, which are supposed to have the capability of automatically selecting relevant information from the input and discarding irrelevant data – by strengthening the links between certain neurons and weakening others during training, step by step, to minimize output error. Another notable difference between the existing literature and the new data is that benchmarks tend to fully rely on the information stored in the word representations and do not work in the absence of these data. Our frame and frame element recognition system learns contextual information from the FrameNet sentences, too, even when large-scale distributional information is not encoded in the representations of the input words.

4 Conclusion

The idea of using small, frame-specific neural networks to carry out semantic processing has been put to the test in this paper using 9 different word representation methods. These frame-specific networks work in their own physical space and attach labels independently of each other. For some applications, e.g. topic detection, this approach may be very useful on its own; for other applications, it may provide important additional information or perhaps a novel, semantically rich starting point for further analysis.

The present approach to semantic frame and frame element labelling has very low resource needs:

- the frame-specific networks are very compact,
- they only need a small set of training examples and counterexamples,
- frame recognition accuracy is high even in the absence of any form of linguistic preprocessing.

We have seen that FrameNet frame labelling benefits from adding information about the distributional features of words. The one-hot representation of words is commonly used in the literature under various names, and it is much easier to create than distributional representations. However, it is clearly inferior to the distributional methods in the semantic labelling task. In neural models, it has another major drawback: it needs as many input neurons as word types, and the introduction of more input neurons increases the number of trainable connections, too, contributing to higher computational complexity. Moreover, since we need to "hard-wire" new input words by adding new nodes and connections to the neurons in the next layer of processing, we also need to do extensive re-training on the entire network to accommodate new vocabulary items. This is not feasible – neither for computation, nor for modelling language acquisition. With distributional representations, the high-level semantic

processor continues to have the same topology, the same number of input nodes and the same number of trainable connections.

The performance of the tested methods has been evaluated quantitatively, also in accordance with the preferences of the computational linguistic community, but a qualitative analysis must also be carried out later to find opportunities for improvement, with an emphasis on frame element recognition. A linguistic perspective also emerges in which unprocessed, unlabelled co-occurrence information directly contributes to describing semantic phenomena. The experiments are also neurolinguistically interpretable: the vehicle of implementation is a neural network for semantic role recognition and also for creating the predict-type word embeddings. In this way, we build a homogeneous, neural-network-based semantic processor that exploits large-scale (frame-independent, linguistically unprocessed and unlabelled) co-occurrence information and a few semantically well-understood training sentences to predict semantic information about unseen sentences.

References

Baker, C.F., Fillmore, C.J. & Lowe, J.B. (1998): The Berkeley FrameNet project. In: *Proceedings of the COLING-ACL*, Montreal, Canada.

Baroni, M., Dinu, G. & Kruszewski, G. (2014): Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In: *Proceedings of ACL 2014*, 238–247.

Bullinaria, J.A. (1995): Modelling lexical decision: Who needs a lexicon? In: Keating,J.G. (ed.): *Neural Computing Research and Applications III (Proceedings of the Fifth Irish Neural Networks Confrence)*. Maynooth, Ireland, 62–69.

Bullinaria, J.A. & Levy, J.P. (2007): Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods* 39, 510–526.

Bullinaria, J.A. & Levy, J.P. (2012): Extracting semantic representations from word co-occurrence statistics: stop-lists, stemming and SVD. *Behavior Research Methods* 44, 890–907.

Elman, J.L. (1990): Finding structure in time. *Cognitive Science* 14, 179–211.

Hermann, K.M., Das, D., Weston, J. & Ganchev, K. (2014): Semantic Frame Identification with Distributed Word Representations. In: *Proceedings of ACL*.

Hodgson, J.M. (1991): Informational constraints on pre-lexical priming. *Language and Cognitive Processes* 6, 169–205.

McClelland, J.L., & Rumelhart, D.E. (1981): An Interactive Activation Model of Context Effects in Letter Perception. Part 1: An Account of Basic Findings. *Psychological Review* 88, 375–407.

Mikolov, T., Chen, K., Corrado, G. & Dean, J. (2013a): Efficient Estimation of Word Representations in Vector Space. Retrieved December 1, 2017, from https://arxiv.org/abs/1301.3781.

Mikolov, T., Sutskever, I., Chen, K, Corrado, G.S. & Dean, J. (2013b): Distributed representations of words and phrases and their compositionality. In: *Advances in neural information processing systems*, 3111–3119.

Pado, S. & Lapata. M. (2007): Dependency-based Construction of Semantic Space Models. *Computational Linguistics* 33:2, 161–199.

Pennacchiotti, M., Cao, D.D., Basili, R., Croce, D. & Roth, M. (2008): Automatic induction of FrameNet lexical units. In: *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP-08)*, 457–465.

Rohde, D.L.T. (1999): *LENS: The light, efficient network simulator. Technical Report CMU-CS-99-164*. Carnegie Mellon University, Department of Computer Science. Pittsburgh, PA.

Scheible, S., Schulte im Walde, S. & Springorum, S. (2013): Uncovering Distributional Differences between Synonyms and Antonyms in a Word Space Model. In: *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, 489–497.

Tóth, Á. (2014): *The Company that Words Keep: Distributional Semantics*. Debrecen: Debrecen University Press.

Dr. Ágoston Tóth
University of Debrecen
Institute of English and American Studies
Pf. 400
H-4002 Debrecen
toth.agoston@arts.unideb.hu