Find the word that does not belong:
A Framework for an Intrinsic Evaluation of Word Vector Representations

José Camacho-Collados and Roberto Navigli
Department of Computer Science
Sapienza University of Rome
{collados,navigli}@di.uniroma1.it

Abstract

We present a new framework for an intrinsic evaluation of word vector representations based on the outlier detection task. This task is intended to test the capability of vector space models to create semantic clusters in the space. We carried out a pilot study building a gold standard dataset and the results revealed two important features: human performance on the task is extremely high compared to the standard word similarity task, and state-of-the-art word embedding models, whose current shortcomings were highlighted as part of the evaluation, still have considerable room for improvement.

1 Introduction

Vector Space Models have been successfully used on many NLP tasks (Turney and Pantel, 2010) such as automatic thesaurus generation (Crouch, 1988; Curran and Moens, 2002), word similarity (Deerwester et al., 1990; Turney et al., 2003; Radinsky et al., 2011) and clustering (Pantel and Lin, 2002), query expansion (Xu and Croft, 1996), information extraction (Laender et al., 2002), semantic role labeling (Erk, 2007; Pennacchiotti et al., 2008), spelling correction (Jones and Martin, 1997), and Word Sense Disambiguation (Navigli, 2012). These models are in the main based on the distributional hypothesis of Harris (1954) claiming that words that occur in the same contexts tend to have similar meanings. Recently, more complex models based on neural networks going beyond simple co-occurrence statistics have been developed (Mikolov et al., 2013; Pennington et al., 2014) and have proved beneficial on key NLP applications such as syntactic parsing (Weiss et al., 2015), Machine Translation (Zou et al., 2013), and Question Answering (Bordes et al., 2014).

Word similarity, which numerically measures the extent to which two words are similar, is generally viewed as the most direct intrinsic evaluation of these word vector representations (Baroni et al., 2014; Levy et al., 2015). Given a gold standard of human-assigned scores, the usual evaluation procedure consists of calculating the correlation between these human similarity scores and scores calculated by the system. While word similarity has been shown to be an interesting task for measuring the semantic coherence of a vector space model, it suffers from various problems. First, the human inter-annotator agreement of standard datasets has been shown to be relatively too low for it to be considered a reliable evaluation benchmark (Batchkarov et al., 2016). In fact, many systems have already surpassed the human inter-annotator agreement upper bound in most of the standard word similarity datasets (Hill et al., 2015). Another drawback of the word similarity evaluation benchmark is its simplicity, as words are simply viewed as points in the vector space. Other interesting properties of vector space models are not directly addressed in the task.

As an alternative we propose the outlier detection task, which tests the capability of vector space models to create semantic clusters (i.e. clusters of semantically similar items). As is the case with word similarity, this task aims at evaluating the semantic coherence of vector space models, but providing two main advantages: (1) it provides a clear gold standard, thanks to the high human performance on the task, and (2) it tests an interesting language understanding property of vector space models not fully addressed to date, and this is their ability to create semantic clusters in the vector space, with potential applications to various NLP tasks.
2 Outlier Detection Task

The proposed task, referred to as **outlier detection** henceforth, is based on a standard vocabulary question of language exams (Richards, 1976). Given a group of words, the goal is to identify the word that does not belong in the group. This question is intended to test the student’s vocabulary understanding and knowledge of the world. For example, *book* would be an outlier for the set of words *apple, banana, lemon, book, orange*, as it is not a fruit like the others. A similar task has already been explored as an *ad-hoc* evaluation of the interpretability of topic models (Chang et al., 2009) and word vector dimensions (Murphy et al., 2012; Fyshe et al., 2015; Faruqui et al., 2015).

In order to deal with the outlier detection task, vector space models should be able to create semantic clusters (i.e., fruits in the example) compact enough to detect all possible outliers. A formalization of the task and its evaluation is presented in Section 2.1 and some potential applications are discussed in Section 2.2.

### 2.1 Formalization

Formally, given a set of words $W = \{w_1, w_2, \ldots, w_n, w_{n+1}\}$, the task consists of identifying the word (outlier) that does not belong to the same group as the remaining words. For notational simplicity, we will assume that $w_1, \ldots, w_n$ belong to the same cluster and $w_{n+1}$ is the outlier. In what follows we explain a procedure for detecting outliers based on semantic similarity.

We define the **compactness score** $c(w)$ of a word $w \in W$ as the compactness of the cluster $W \setminus \{w\}$, calculated by averaging all pair-wise semantic similarities of the words in $W \setminus \{w\}$:

$$c(w) = \frac{1}{k} \sum_{w_i \in W \setminus \{w\}} \sum_{w_j \in W \setminus \{w\}} \text{sim}(w_i, w_j)$$

where $k = n(n-1)$. We propose two measures for computing the reliability of a system in detecting an outlier given a set of words: Outlier Position (OP) and Outlier Detection (OD). Given a set $W$ of $n+1$ words, OP is defined as the position of the outlier $w_{n+1}$ according to the compactness score, which ranges from 0 to $n$ (position 0 indicates the lowest overall score among all words in $W$, and position $n$ indicates the highest overall score). OD is, instead, defined as 1 if the outlier is correctly detected (i.e. $OP(w_{n+1}) = n$) and 0 otherwise.

To estimate the overall performance on a dataset $D$ (composed of $|D|$ sets of words), we define the Outlier Position Percentage (OPP) and Accuracy measures:

$$\text{OPP} = \frac{\sum_{W \in D} OP(W)}{|D|} \times 100$$

$$\text{Accuracy} = \frac{\sum_{W \in D} OD(W)}{|D|} \times 100$$

The **compactness score** of a word may be expensive to calculate if the number of elements in the cluster is large. In fact, the complexity of calculating $OP$ and $OD$ measures given a cluster and an outlier is $(n + 1) \times n \times (n - 1) = O(n^3)$. However, this complexity can be effectively reduced to $(n + 1) \times 2n = O(n^2)$. Our proposed calculations and the proof are included in Appendix A.

### 2.2 Potential applications

In this work we focus on the intrinsic semantic properties of vector space models which can be inferred from the outlier detection task. In addition, since it is a task based partially on semantic similarity, high-performing models in the outlier detection task are expected to contribute to applications in which semantic similarity has already shown its potential: Information Retrieval (Hliaoutakis et al., 2006), Machine Translation (Lavie and Denkowski, 2009), Lexical Substitution (McCarthy and Navigli, 2009), Question Answering (Mohler et al., 2011), Text Summarization (Mohammad and Hirst, 2012), and Word Sense Disambiguation (Patwardhan et al., 2003), to name a few. Furthermore, there are other NLP applications directly connected with the semantic clustering proposed in the outlier detection task. Ontology Learning is probably the most straightforward application, as a meaningful cluster of items is expected to share a common hypernym, a property that has already been exploited in recent studies using embeddings (Fu et al., 2014; Espinosa-Anke et al., 2016). In fact, building ontologies is a time-consuming task and generally relies on automatic or semi-automatic steps (Velardi et al., 2013; Alfarone and Davis, 2015). Ontologies are one of the basic components of the Semantic Web (Berners-Lee et al., 2000) and have already proved their importance in downstream applications like Question Answering (Mann, 2002),
which in the main rely on large structured knowledge bases (Bordes et al., 2014).

In this paper we do not perform any quantitative evaluation to measure the correlation between the performance of word vectors on the outlier detection task and downstream applications. We argue that the conclusions drawn by recent works (Tsvetkov et al., 2015; Chiu et al., 2016) as a result of measuring the correlation between standard intrinsic evaluation benchmarks (e.g. word similarity datasets) and downstream task performances are hampered by a serious methodological issue: in both cases, the sample set of word vectors used for measuring the correlation is not representative enough, which is essential for this type of statistical study (Patton, 2005). All sample vectors came from corpus-based models\(^1\) trained on the same corpus and all perform well on the considered intrinsic tasks, which constitute a highly homogeneous and not representative sample set. Moreover, using only a reduced selected set of applications does not seem sufficient to draw general conclusions about the quality of an intrinsic task, but rather about its potential on those specific applications. Further work should focus on these issues before using downstream applications to measure the impact of intrinsic tasks for evaluating the quality of word vectors. However, this is out of the scope of this paper.

### Table 1: First four clusters (including outliers) of the 8-8-8 outlier detection dataset.

| Cluster elements | Big cats | European football teams | Solar System planets | Months |
|------------------|----------|------------------------|----------------------|--------|
| tiger            | FC Barcelona | Mercury               | January              |        |
| lion             | Bayern Munich | Venus                  | March                |        |
| cougar           | Real Madrid | Earth                  | May                  |        |
| jaguar           | AC Milan  | Mars                   | July                 |        |
| leopard          | Juventus   | Jupiter                | September            |        |
| cheetah          | Atletico Madrid | Saturn              | November             |        |
| wildcat          | Chelsea    | Uranus                 | February             |        |
| lynx             | Borussia Dortmund | Neptune            | June                 |        |
| 1st Outlier      | dog       | Miami Dolphins         | Sun                  | Wednesday |
| 2nd Outlier      | mouse     | McLaren                | Moon                 | winter  |
| 3rd Outlier      | dolphin   | Los Angeles Lakers     | Triton               | date   |
| 4th Outlier      | shark     | Bundesliga             | Comet Halley         | year   |
| 5th Outlier      | savanna   | football               | eclipse              | astrology |
| 6th Outlier      | jungle    | goal                   | astronaut            | birthday |
| 7th Outlier      | day       | couch                  | lunch                | ball   |
| 8th Outlier      | car       | fridge                 | window               | paper  |

3 Pilot Study

We carried out a pilot study on the outlier detection task. To this end, we developed a new dataset, 8-8-8 henceforth. The dataset consisted of eight different topics each made up of a cluster of eight words and eight possible outliers. Four annotators were used for the creation of the dataset. Each annotator was asked to first identify two topics, and for each topic to provide a set of eight words belonging to the chosen topic (elements in the cluster), and a set of eight heterogeneous outliers, selected varying their similarity to and relatedness with the elements of the cluster\(^2\). In total, the dataset included sixty-four sets of 8 + 1 words for the evaluation. Tables 1 and 2 show the eight clusters and their respective outliers of the 8-8-8 outlier detection dataset.

When we consider the time annotators had to spend creating the relatively small dataset for this pilot study, the indications are that building a large-scale dataset may not need to be very time-consuming. In our study, the annotators spent most of their time reading and understanding the guidelines, and then thinking about suitable topics. In fact, with a view to constructing a large-scale dataset, this topic selection step may be carried out prior to giving the assignments to the annotators, providing topics to annotators according to their

---

\(^1\)In the case of Chiu et al. (2016) all word vectors in the sample come from the Skip-Gram model of Word2Vec (Mikolov et al., 2013).

\(^2\)We release the full dataset and guidelines for the creation of the topics at [http://lcl.uniroma1.it/outlier-detection](http://lcl.uniroma1.it/outlier-detection)
expertise. The time spent for the actual creation of a cluster (including outliers) was in all cases less than ten minutes.

### 3.1 Human performance

We assessed the human performance of eight annotators in the task via accuracy. To this end, each annotator was given eight different groups of words, one for each of the topics of the 8-8-8 dataset. Each group of words was made up of the set of eight words comprising the cluster, plus one additional outlier. All the words were shuffled and given to the annotator without any additional information (e.g. annotators did not know the topic of the cluster). The task for the annotators consisted of detecting the outlier in each set of nine words. To this end, each annotator was asked to provide two different answers: one without any external help, and a second one in which the annotator could use the Web as external help for three minutes before giving his answer. This human performance in the outlier detection task may be viewed as equivalent to the inter-annotator agreement in word similarity, which is used to measure the human performance in the task.

The results of the experiment were the following: an accuracy of 98.4% for the first task in which annotators did not use any external help, and an accuracy of 100% for the second task in which annotators were allowed to use external help. This contrasts with the evaluation performed in word similarity, which is based on human-assigned scores with a relatively low inter-annotator agreement. For example, the inter-annotator agreements in the standard WordSim-353 (Finkelstein et al., 2002) and SimLex-999 (Hill et al., 2015) word similarity datasets were, respectively, 0.61 and 0.67 according to average pair-wise Spearman correlation. In fact, both upper-bound values have already been surpassed by automatic models (Huang et al., 2012; Wieting et al., 2015).

### 3.2 Word embeddings performance

We tested the performance of three standard word embedding models in the outlier detection task: the CBOW and Skip-Gram models of Word2Vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014). We report the results of each of the models trained on the 3B-words UMBC webbase corpus \(^3\) (Han et al., 2013), and the 1.7B-words English Wikipedia \(^4\) with standard hyperparameters. For each of the models, we used as multi-word expressions the phrases contained in the pre-trained Word2Vec word embeddings trained on the Google News corpus. The evaluation was performed as explained in Section 2.1, using cosine

\[^3\]http://ebiquity.umbc.edu/blogger/2013/05/01/umbc-webbase-corpus-of-3b-english-words/

\[^4\]We used the Wikipedia dump of November 2014.

\[^5\]The dimensionality of the vectors was set to 300 for the three models. Context-size 5 for CBOW and 10 for Skip-Gram and GloVe; hierarchical softmax for CBOW and negative sampling for Skip-Gram and GloVe.

---

**Table 2:** Last four clusters (including outliers) from the 8-8-8 outlier detection dataset.

| Cluster elements | IT companies | German car manufacturers | Apostles of Jesus Christ | South American countries |
|------------------|--------------|--------------------------|--------------------------|--------------------------|
|                  | Apple        | Mercedes Benz            | Peter                    | Brazil                   |
|                  | Foxconn      | BMW                      | Andrew                   | Colombia                 |
|                  | Amazon       | Audi                     | James                    | Argentina                |
|                  | HP           | Opel                     | John                     | Peru                     |
|                  | Microsoft    | Volkswagen               | Thaddaeus                | Venezuela                |
|                  | IBM          | Porsche                  | Bartholomew              | Chile                    |
|                  | Google       | Alpina                   | Thomas                   | Ecuador                  |
|                  | Sony         | Smart                    | Matthew                  | Bolivia                  |
| 1st Outlier      | Opel         | Michelin                 | Noah                     | Bogotá                   |
| 2nd Outlier      | Boeing       | Bridgestone              | Mary                     | Rio de Janeiro           |
| 3rd Outlier      | Nestlé       | Boeing                   | Pope Benedict XVI        | New York                 |
| 4th Outlier      | Adidas       | Samsung                  | Ambrose                  | Madrid                   |
| 5th Outlier      | computer     | Michael Schumacher       | crucifixion              | town                     |
| 6th Outlier      | software     | Angela Merkel            | church                   | government               |
| 7th Outlier      | chair         | Capri                    | airplane                 | bottle                   |
| 8th Outlier      | plant         | pineapple                | Microsoft                | telephone                |
Table 3: Outlier Position Percentage (OPP) and Accuracy (Acc.) of different word embedding models on the 8-8-8 outlier detection dataset.

| Model     | Corpus      | OPP | Acc. |
|-----------|-------------|-----|------|
| CBOW      | UMBC        | 93.8| 73.4 |
|           | Wikipedia   | 95.3| 73.4 |
| Skip-Gram | UMBC        | 92.6| 64.1 |
|           | Wikipedia   | 93.8| 70.3 |
|           | Google News | 94.7| 70.3 |
| GloVe     | UMBC        | 81.6| 40.6 |
|           | Wikipedia   | 91.8| 56.3 |

to miss a representation for a given lexicalization if that lexicalization is not found enough times in the corpus. In order to overcome these ambiguity and synonymy issues, it might be interesting for future work to leverage vector representations constructed from large lexical resources such, as FreeBase (Bordes et al., 2011; Bordes et al., 2014), Wikipedia (Camacho-Collados et al., 2015a), or BabelNet (Iacobacci et al., 2015; Camacho-Collados et al., 2015b).

4 Conclusion

In this paper we presented the outlier detection task and a framework for an intrinsic evaluation of word vector space models. The task is intended to test interesting semantic properties of vector space models not fully addressed to date. As shown in our pilot study, state-of-the-art word embeddings perform reasonably well in the task but are still far from human performance. As opposed to the word similarity task, the outlier detection task achieves a very high human performance, proving the reliability of the gold standard. Finally, we release the 8-8-8 outlier detection dataset and the guidelines given to the annotators as part of the pilot study, and an easy-to-use Python code for evaluating the performance of word vector representations given a gold standard dataset at http://lcl.uniroma1.it/outlier-detection.

Acknowledgments

The authors gratefully acknowledge the support of the ERC Starting Grant MultiJEDI No. 259234.

We would like to thank Claudio Delli Bovi, Ilenia Giambruno, Ignacio Iacobacci, Massimiliano Mancini, Tommaso Pasini, Taher Pilehvar, and Alessandro Raganato for their help in the construction and evaluation of the outlier detection dataset. We would also like to thank Jim McManus for his comments on the manuscript.

References

Daniele Alfarone and Jesse Davis. 2015. Unsupervised learning of an is-a taxonomy from a limited domain-specific corpus. In Proceedings of IJCAI.

6This last issue was not present in this evaluation as for the multiword instances we carefully selected the lexicalizations which were covered by the pre-trained Word2Vec vectors, which ensured a full coverage of all models.
Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don’t count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In Proceedings of ACL, pages 238–247.

Miroslav Batchkarov, Thomas Kober, Jeremy Reffin, Julie Weeds, and David Weir. 2016. A critique of word similarity as a method for evaluating distributional semantic models. In Proceedings of the ACL Workshop on Evaluating Vector Space Representations for NLP, Berlin, Germany.

Tim Berners-Lee, Mark Fischetti, and Michael L. Fouad. 2000. Weaving the Web: The original design and ultimate destiny of the World Wide Web by its inventor. HarperCollins.

Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. 2011. Learning structured embeddings of knowledge bases. In Twenty-Fifth AAAI Conference on Artificial Intelligence.

Antoine Bordes, Sumit Chopra, and Jason Weston. 2014. Question answering with subgraph embeddings. In EMNLP.

José Camacho-Collados, Mohammad Taher Pilehvar, and Roberto Navigli. 2015a. NASARI: a Novel Approach to a Semantically-Aware Representation of Items. In Proceedings of NAACL, pages 567–577.

José Camacho-Collados, Mohammad Taher Pilehvar, and Roberto Navigli. 2015b. A Unified Multilingual Semantic Representation of Concepts. In Proceedings of ACL, pages 741–751.

Jonathan Chang, Sean Gerrish, Chong Wang, Jordan L Boyd-Graber, and David M Blei. 2009. Reading tea leaves: How humans interpret topic models. In Advances in neural information processing systems, pages 288–296.

Billy Chiu, Anna Korhonen, and Sampo Pyysalo. 2016. Intrinsic evaluation of word vectors fails to predict extrinsic performance. In Proceedings of the ACL Workshop on Evaluating Vector Space Representations for NLP, Berlin, Germany.

C. J. Crouch. 1988. A cluster-based approach to thesaurus construction. In Proceedings of the 11th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR ’88, pages 309–320.

James R. Curran and Marc Moens. 2002. Improvements in automatic thesaurus extraction. In Proceedings of the ACL-02 Workshop on Unsupervised Lexical Acquisition - Volume 9, ULA ’02, pages 59–66.

Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. 1990. Indexing by latent semantic analysis. Journal of American Society for Information Science, 41(6):391–407.

Katrin Erk. 2007. A simple, similarity-based model for selectional preferences. In Proceedings of ACL, Prague, Czech Republic.

Luís Espinosa-Anke, Horacio Saggion, Francesco Ranzano, and Roberto Navigli. 2016. ExTaSem! Extending, Taxonomizing and Semantifying Domain Terminologies. In Proceedings of the 30th Conference on Artificial Intelligence (AAAI’16).

Manaal Faruqui, Yulia Tsvetkov, Dani Yogatama, Chris Dyer, and Noah Smith. 2015. Sparse overcomplete vector representations. In Proceedings of ACL, Beijing, China.

Lev Finkelstein, Gabrilovich Evgeniy, Matias Yossi, Rivlin Ehud, Solan Zach, Wolfman Gadi, and Ruppin Eytan. 2002. Placing search in context: The concept revisited. ACM Transactions on Information Systems, 20(1):116–131.

Ruiji Fu, Jiang Guo, Bing Qin, Wanxiang Che, Haifeng Wang, and Ting Liu. 2014. Learning semantic hierarchies via word embeddings. In ACL (1), pages 1199–1209.

Alona Fyshe, Leila Wehbe, Partha P Talukdar, Brian Murphy, and Tom M Mitchell. 2015. A compositional and interpretable semantic space. In Proc. of NAACL.

Lushan Han, Abhay Kashyap, Tim Finin, James Mayfield, and Jonathan Weese. 2013. UMBC EBIQUITY-CORE: Semantic textual similarity systems. In Proceedings of the Second Joint Conference on Lexical and Computational Semantics, volume 1, pages 44–52.

Zellig Harris. 1954. Distributional structure. Word, 10:146–162.

Felix Hill, Roi Reichart, and Anna Korhonen. 2015. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. Computational Linguistics.

Angelos Hliaoutakis, Giannis Varelas, Epimenidis Voutsakis, Euripides GM Petrakis, and Evangelos Milios. 2006. Information retrieval by semantic similarity. International Journal on Semantic Web and Information Systems, 2(3):55–73.

Eric H. Huang, Richard Socher, Christopher D. Manning, and Andrew Y. Ng. 2012. Improving word representations via global context and multiple word prototypes. In Proceedings of ACL, pages 873–882, Jeju Island, Korea.

Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. 2015. Sensembled: Learning sense embeddings for word and relational similarity. In Proceedings of ACL, pages 95–105, Beijing, China.

Michael P. Jones and James H. Martin. 1997. Contextual spelling correction using latent semantic analysis. In Proceedings of the Fifth Conference on
A Proposition 1

The complexity for calculating \( OP(w) \) can be reduced to \( 2n \) by calculating the following pseudo-inverted compactness score\(^7\) \( p(w) \) instead of the compactness score \( c(w) \) of Equation 1, and defining \( OP_p(w) \) as the position of the outlier in \( W \) according to the inverted pseudo-inverted compactness score:

\[
p(w) = \frac{1}{k'} \left( \sum_{w_i \in W \atop w_i \neq w} \text{sim}(w_i, w) + \sum_{w_i \in W \atop w_i \neq w} \text{sim}(w, w_i) \right)
\]

(4)

where \( k' = 2(|W| - 1) \).

Proof. Since \( OP(w) \) is given by the position of \( c(w) \) with respect to the remaining words in \( W \) and \( \leq \) represents a relation of total order, we only have to prove the following statement:

\[
c(w) \leq c(w') \iff p(w') \leq p(w), \forall w, w' \in W
\]

(5)

Given any \( w \in W \), we can calculate the sum of all pair-wise similarities of the words in \( W \) (i.e. \( \mu \)) as follows:

\[
\mu = \sum_{w_i \in W \setminus \{w\}} \sum_{w_j \in W \setminus \{w\} \atop w_j \neq w_i} \text{sim}(w_i, w_j) + \\
+ \sum_{w_i \in W \setminus \{w\}} \text{sim}(w_i, w) + \sum_{w_i \in W \setminus \{w\}} \text{sim}(w, w_i) = k \cdot c(w) + k' \cdot p(w)
\]

(6)

where \( k = (|W| - 1)(|W| - 2) \). Therefore,

\[
\mu = k \cdot c(w) + k' \cdot p(w), \forall w \in W
\]

(7)

Since \( k, k' \) (being both \( k \) and \( k' \) positive values) and \( \mu \) are all fixed values only depending on \( W \), we can trivially infer the following statement from Equation 7 given any \( w, w' \in W \):

\[
c(w) \leq c(w') \iff p(w') \leq p(w)
\]

(8)

\(^7\)In this proposition we do not assume any special property to the function \( \text{sim}(\ldots) \) for generalization. If \( \text{sim}(\ldots) \) were symmetrical (e.g. cosine similarity is symmetrical), we could simply define the pseudo-inverted compactness score as \( p(w) = \sum_{w_i \in W} \text{sim}(w_i, w) \), which would lead to a complexity of \( n \).