Modelling Word Associations with Word Embeddings for a Guesser Agent in the Taboo City Challenge Competition

Verna Dankers, Aysenur Bilgin, Raquel Fernández

Institute for Logic, Language and Computation, University of Amsterdam
vernadankers@gmail.com, {a.bilgin|raquel.fernandez}@uva.nl

Abstract In the Taboo City Challenge, artificial agents should guess the names of cities from simple textual hints and are evaluated with games played by humans. Thus, playing the games successfully requires mimicking associations that humans have with geographical locations. In this paper, an architecture is proposed that calculates the associative similarity between a city and a hint from a semantic vector space. The semantic vector space is created using the Skip-gram hierarchical softmax model, from a tailored corpus about travel destinations. We investigate the effect of varying training parameters and introduce a targeted corpus annotation method that significantly improves performance. The results on a dataset of 149 games indicate that the proposed architecture can guess the target city with up to 22.45% accuracy — a substantial improvement over the 4.11% accuracy achieved by the baseline architecture.

1 Introduction

Taboo is a word-guessing game, in which one agent provides clues about the term to be guessed without mentioning the target term or any of the related terms from a list of taboo words. Thus, the game requires the describer to think of well-known facts about the target word to enable the other agent to successfully guess it. The Taboo City Challenge[1] is a competition inspired by Taboo, where artificial guesser agents play the Location Taboo Game (LTG). The objective of the LTG is to guess the names of cities from simple textual hints. The data provided for training and testing the artificial guesser agent (AGA) are games that were successfully played by various human players. To achieve the optimal game score, the AGA should minimise the number of hints used before the term is correctly guessed. Therefore, the AGA should be able to mimic word associations that human players have with geographical locations.

Word associations have been obtained in multiple ways in the literature: through free association norms (De Deyne and Storms 2008), through semantic networks, and by inferring them from corpora (Heath et al. 2013). The latter

[1] The challenge is organised by the ESSENCE Network https://www.essence-network.com/challenge/.

approach uses distributional semantic models (DSM) that learn word embeddings, represented as real-valued vectors, from patterns of word co-occurrences in a corpus. These models rely on the Distributional Hypothesis, according to which words that appear in similar contexts tend to have related meanings (Harris 1954). DSMs have mostly been applied to semantic similarity tasks rather than to associative similarity tasks (Turney and Pantel 2010), but they can be used to mirror word associations made by humans as well (Peirsman et al. 2008). While the gathering of free association norms tends to be expensive and time-consuming for a domain-specific task such as the LTG, the extraction of associations from a corpus through a DSM is fully data-driven and automatic.

The AGA presented in this paper employs a DSM, the Skip-gram embedding model introduced by Mikolov et al. (2013a), to create word representations from a tailored corpus, which is constructed with data from the online encyclopedias Wikipedia[2] and Wikivoyage.[3] These sources, based on the knowledge and experience of the volunteer authors, implicitly contain associations humans have with geographical locations.

The rest of the paper is organised as follows: In the following section related work is reviewed. Section 3 discusses the rules of the LTG. Section 4 presents the proposed AGA architecture and discusses the algorithms used. The experiments and results are presented in Section 5. The results are discussed in Section 6 and some concluding remarks and future research directions are given in Section 7.

2 Related Work

Heath et al. (2013) reviewed methods for building an agent for a word-guessing game named Wordlery, which is similar to Taboo. In their work, two methods for obtaining word associations were considered: using human free association norms and applying count-based semantic models to build word vectors from a corpus. The models were evaluated by playing the game. The free association norms outperformed the count-based semantic models. Combining the two methods of forming word associations was superior to each of the methods in isolation. Heath et al. (2013) suggested that more advanced corpus-based semantic models, which take into account additional semantic information, may improve the results on similar tasks. In this work, we build upon this suggestion and propose a novel corpus annotation method to improve corpus-based semantic models.

The baseline for the performance of AGAs has been set by Adrian et al. (2016), who proposed a semantic distance based architecture. The architecture uses a two-step approach. First, the geographical area of the guess is narrowed down to the country. Next, the area is further narrowed down to the city. Two

[2] Wikipedia is an online collaborative encyclopedia. The dump used was from April 20, 2017, https://dumps.wikimedia.org/enwiki/.

[3] Wikivoyage is a global travel guide for travel destinations and travel topics written by volunteer authors. The dump used was from May 1, 2017, https://dumps.wikimedia.org/enwikivoyage/.

types of resources are used to measure the distance between a geographical location and a hint: WordNet and Wikipedia. Wordnet is used to measure semantic distance through the hierarchical relations of the location and the hint. For Wikipedia, the similarity is measured by combining the number of hits for the hint, the location and the combination of both, through multiple association metrics. The highest score, yielding 23.17% accuracy and 68.42% faster guessing performance, was achieved with the Wikipedia corpus and the Pointwise Mutual Information metric. A quantitative comparison of this baseline and the AGA proposed in this paper, is provided in Section 5.

3 Location Taboo Game

In this section, the rules for the LTG are laid out.[4] An LTG is played by a *describer* agent and a *guesser* agent. Hints are simple English noun phrases, consisting of one to three words that are common nouns, adjectives, or connectors. The hints may not include proper nouns. For example, if *'Verona'* is the target city, the clue *'Romeo and Juliet'* is not allowed, but *'tragic love story'* is. Although there is no closed set of cities available, the challenge does focus on well-known cities.

The describer starts the game by providing a hint about the target city. Based on this hint, the guesser tries to guess the city that is being described. As long as the guess is incorrect, the describer provides a new hint and the game continues until there are no more hints left. If the guesser has not been able to find the right city before the describer runs out of hints, the game is considered to have failed. The AGA architecture is required to minimise the total number of guesses needed before the target city has been found.

In the competition, the clues from the describer agent are hints from real games played by human players, for which the target city was guessed successfully. Therefore, the number of hints available differs per game, depending on how many clues the human player needed. 149 real-world games were made available by the ESSENCE Network through an API.[5] The number of hints per game varies from 1 to 10 and the average number of hints is 2.7. An important difference with a real, interactive word-guessing game is that the hints from the API games are static; they do not depend upon the guesses that are given.

For evaluating the AGA performance, the total game score that is derived by the number of submitted guesses is used:

$$(5 \cdot f + \sum_{i=0}^{j} h_i) \tag{1}$$

where j represents the number of games for which the guesser agent is evaluated, h_i the number of hints used in game i and f is the number of failed games.

[4] Visit https://www.essence-network.com/challenge/ for a complete specification.

[5] The API: http://challenge.essence-network.com/describer/v1/. The results are based on the games from the API from June 1, 2017 and June 26, 2017.

4 Proposed Guesser Agent Architecture

4.1 Data

Preprocessing We used data sets of Wikipedia and Wikivoyage, available under an open license by the Wikimedia Foundation. An advantage of Wikivoyage is that all entries come with information that is useful for tourists. However, many of these pages are not specifically about travel destinations but rather about tourist attractions, such as historic buildings or tours. Therefore, an initial filtering of the entries of Wikipedia and Wikivoyage was performed using the database of populated places from Natural Earth cultural vectors.[6] This database includes all capitals, major cities and towns, and smaller towns from sparsely inhabited regions. The result of this initial filtering was a set of 215 countries and 7,267 cities and towns. The Wikipedia and Wikivoyage pages entitled with these names were extracted. For the Wikivoyage pages, the outlinks to other Wikivoyage pages were considered relevant and were included in the corpus. A second filter was applied to the resulting text corpus to remove markup language and English stop words. The corpus was split into sentences before punctuation was removed. City names composed of two or more words were joined: *'New York'* is represented as *'New_York'*. All tokens were lowercased apart from the city names to avoid bias in the word vectors of city names that are also known as common nouns, such as *'Tours'*. The final normalised corpus consisted of around 17 million tokens, and contained 600,000 types.

Proposed Targeted Annotation In order to improve the association between a geographical location and its corresponding Wikipedia and Wikivoyage pages, we introduce a novel type of targeted annotation. For the pages entitled with the name of a country or city, this name is inserted in every sentence. The number of insertions is calculated by rounding up the length of the sentence divided by 50.

Candidate Cities A large part of the corpus incorporates little-known locations that are not likely to appear in the Taboo City Challenge. Considering all cities in the world could result in unexpected results due to data sparseness. Therefore, candidate cities were filtered in order to only consider sufficiently relevant places. This filter uses a popularity metric based on the ranking of cities on the popular website NomadList,[7] where travel destinations can be ranked based on multiple factors. The top 400 cities, ranked according to the number of visits, were used for the list of candidate cities.

4.2 Model

Semantic Vector Space Mikolov et al. (2013a) proposed two word-embedding models known as `word2vec`: Skip-gram (SG) and Continuous Bag of Words

[6] http://www.naturalearthdata.com/downloads/10m-cultural-vectors/

[7] https://nomadlist.com/

(CBOW). Within the SG algorithm the goal is to set the word embedding parameters so as to maximise the probability of the context, given the term that is in the centre. The training objective of CBOW, on the other hand, is to maximise the probability of the centre word based on its context. Multiple variants of the SG and CBOW models were considered for the creation of the semantic vector space.

Two extensions to the original algorithms, proposed by Mikolov et al. (2013b), are the application of hierarchical softmax (HS) and negative sampling (NS). While the basic formulations of the algorithms use the full softmax function, HS approximates full softmax efficiently by using a binary Huffman tree for the representation of the output layer. The basic algorithms update all output vectors per iteration. NS only updates the vector for the output word, as a positive example, and samples n words as negative samples, where n is a parameter.

Similarity Measure Our AGA uses cosine similarity to calculate the level of association between the word vector of a hint and the word vector of a city.

Game Strategy The AGA retrieves the cities nearest to the hints in the semantic vector space via cosine similarity and employs a game strategy to choose from these cities. We considered multiple game strategies: two strategies that are the adapted versions of cross situational learning algorithms named Enumeration and Elimination (De Beule 2016), a two-step strategy similar to the approach of Adrian et al. (2016), and a strategy based on vector arithmetic. Due to space limitations, only the results for the strategy employed in the final architecture, Enumeration, are presented here. The Enumeration game strategy uses the cumulative cosine similarity scores for hints and cities and guesses the city with the highest score.

4.3 Implementation

The AGA architecture was implemented in Python 2.7.[8] For the processing of the corpus, the Python library `gensim.corpora.wikicorpus`[9] was used for the removal of markup language. The list of English stopwords used was the one provided with the NLTK Toolkit.[10] The SG and CBOW models as implemented in `gensim.models.word2vec`[11] were used to train the word embeddings.

The main AGA architecture as used for the competition is shown in Algorithm 1. If the first hint is not present in the semantic vector space, a default set of 300-dimensional SGNS vectors trained on a Google News dataset[12] is used.

[8] https://github.com/vdankers/LocationTabooAgents

[9] The library used to construct a corpus from a MediaWiki-based database dump is https://radimrehurek.com/gensim/corpora/wikicorpus.html.

[10] http://www.nltk.org/

[11] Python implementation of `word2vec`: https://radimrehurek.com/gensim/models/word2vec.html.

[12] https://code.google.com/archive/p/word2vec/

Algorithm 1: Main algorithm of guesser agent

```
Input : candidate_cities, wiki_vectors, google_news_vectors
while guessed = false and new_hints_exist = true do
    hint ⟵ getHint();
    if hint not in wiki_vectors and empty(all_hints) then
        vectors ⟵ google_news_vectors
    end
    else
        vectors ⟵ wiki_vectors
    end
    all_hints ⟵ addHint(hint);
    for city in candidate_cities do
        city.score ⟵ cumulativeCosineSimilarity(all_hints, vectors)
    end
    guess_of_agent ← getHighestScore(candidate_cities)
    candidate_cities ⟵ remove(guess_of_agent)
    guessed ⟵ guessCity(guess_of_agent)
    new_hints_exist ⟵ askAPI()
end
```

In the games from the API there were six games for which the similarity scores for the first hint had to be extracted from the Google News vectors.

5 Experiments and Results

5.1 Experimental Setup

Several variants of the SG model and the CBOW model were trained with different parameters to find the best configuration for the Taboo City Challenge. The number of training epochs was set to 15. For the negative sampling algorithm 5 negative samples were used. The settings that are varied in the results presented in Section 5.2 are the vector size and the context window size. We consider vectors of 200, 300 and 400 dimensions and context window sizes of 5, 10, 15, 20 and 25 words. A variety of metrics can be used to calculate the similarity between two vectors; here, we consider the cosine similarity measure.

The top 400 cities from NomadList and the 149 games from the ESSENCE Network API were used to test the performance of the proposed AGA and the semantic distance based architecture of Adrian et al. (2016). To assess the suitability of the semantic vector space created for the proposed AGA, we also test the performance of the AGA with a pre-trained vector space: the SGNS vectors trained on the Google News dataset.

5.2 Results

We evaluate the AGA architecture according to three metrics: the game score, the accuracy and the faster guessing performance (FGP). The FGP represents

Table 1: The game score, accuracy and FGP for the top 5 best configurations per algorithm. The FGP represents the percentage of successfully finished games for which the AGA found the target city faster than the human player.

(a) Unannotated corpus

Model	`dim`	`win`	Score	Accuracy (%)	FGP (%)
SGHS	300	20	**971**	**19.05**	32.14
	200	25	984	17.69	**46.15**
	400	20	991	17.01	44.00
	300	25	994	17.01	44.00
	400	25	999	16.33	45.83
SGNS	200	20	**998**	**15.65**	**47.83**
	400	20	1006	**15.65**	26.09
	300	25	1016	14.29	38.10
	200	5	1021	14.29	28.57
	200	15	1026	13.61	40.00
CBOW HS	200	15	**1025**	**12.93**	36.84
	200	20	1034	11.56	52.94
	200	25	1044	11.56	29.41
	300	20	1050	10.20	26.67
	300	25	1051	10.20	**53.33**
CBOW NS	300	25	**1024**	**12.24**	50.00
	400	25	1029	11.56	52.94
	400	15	1031	10.88	**68.75**
	400	20	1037	10.88	50.00
	200	25	1039	10.88	50.00

(b) Annotated corpus

Model	`dim`	`win`	Score	Accuracy (%)	FGP (%)
SGHS	400	15	**935**	**22.45**	**45.45**
	400	25	946	21.09	45.16
	400	10	947	22.45	27.27
	300	25	949	21.77	37.50
	200	10	952	21.09	45.16
SGNS	300	25	**964**	**21.09**	29.03
	300	20	969	20.41	30.00
	200	25	966	20.41	36.67
	200	10	972	19.05	46.43
	200	20	978	17.69	**50.00**
CBOW HS	300	5	**1000**	**16.33**	45.83
	400	5	**1000**	**16.33**	45.83
	200	10	1001	15.65	52.17
	400	10	1013	14.29	**61.90**
	200	5	1022	13.61	45.00
CBOW NS	400	20	**997**	**15.65**	43.48
	300	20	1001	**15.65**	43.48
	200	10	1001	14.97	**54.55**
	400	15	1002	15.65	30.43
	400	10	1003	14.97	50.00

the percentage of successfully finished games for which the AGA found the target city faster than the human player. The results of the top 5 best-scoring configurations based on the game score are displayed in Table 1 for both the unannotated corpus and the annotated corpus. The baseline AGA had a game score of 1101, an accuracy of 4.11% and an FGP of 33.33%. The pre-trained vectors gave a game score of 982, an accuracy of 18.37% and an FGP of 40.74%. The targeted corpus annotation improved performance significantly for SGHS, SGNS and CBOWNS (two-sided relative t-test, $p < 0.001$).

6 Discussion

The different configurations used in the experiments demonstrate the optimal settings for neural word embeddings for the Taboo City Challenge. A rather large context window size of 25 yields the best results on average. The fact that a larger context window size is more suitable for capturing associative similarity is consistent with the findings of Peirsman et al. (2008), who show that large context windows tend to model human associations better.

A targeted annotation method has been proposed to make the association between a country or city and the content of its page more explicit during the construction of word embeddings. For the pages entitled with the name of a country or city, this name was inserted in every sentence. Human readers or authors implicitly assume this association and the annotation makes the association explicit for the DSM. The application of the proposed annotation significantly improves the results. When comparing the top configurations per algorithm for both corpora, the annotation gives 3.40% absolute improvement for SGHS, 5.44% for SGNS, 3.40% for CBOWHS, 3.41% for CBOWNS.

The difference in performance between SG and CBOW may be explained by the size of the corpus and the models' training objectives. The results indicate that SG performs better at the LTG than CBOW, and that HS is more suitable for the game than NS. An explanation for the difference between the scores of HS and NS is that more frequent words are more likely to be selected as negative samples. Therefore, NS represents frequent words better than infrequent words. Hints can be very infrequent words, as they are specific for a certain target city.

The choice for a tailored corpus for the AGA architecture is based upon the assumption that the associations with cities depend upon their role as travel destination (Wikivoyage) or general facts about places (Wikipedia). Corpora from other domains, for example news items, may enhance the associations. A possible disadvantage of the resources used may be that encyclopedias are mostly useful for little-known facts, whereas the associations that play a role in word-guessing games may be significantly simpler (Von Ahn et al. 2006).

7 Conclusion and Future Work

In this work, we propose an AGA architecture that uses context-predicting DSMs to infer word associations from a tailored corpus, for which we propose a targeted annotation that improves the performance significantly. Based on the results, the best settings for the AGA architecture are the SGHS model, a context window size of 15 and a vector size of 400. The architecture is an improvement compared to the baseline architecture of Adrian et al. (2016) and can guess target cities from the set of games provided by the ESSENCE Network with up to 22.45% accuracy. Regarding corpus improvements, future work can involve combining multiple types of resources, for example reviews from tourists and news articles about a city. Secondly, future research could focus on defining a metric that better expresses which cities are well-known and thus more likely to appear in the LTG. Thirdly, more complex architectures could be created. A better method could be developed for the interpretation of multi word hints and the knowledge of multiple agents could be used in a larger system in which agents cooperate to find the target city.

References

Adrian, K., Bilgin, A., Van Eecke, P. A Semantic Distance based Architecture for a Guesser Agent in ESSENCEs Location Taboo Challenge. DIVERSITY@ ECAI 2016, 33–39 (2016)

De Pessemier, T. Dhondt, J., Vanhecke, K., Martens, L. TravelWithFriends: a hybrid group recommender system for travel destinations. In: Workshop on Tourism Recommender Systems (TouRS15), in conjunction with the 9th ACM Conference on Recommender Systems (RecSys 2015), 51–60 (2015)

Baroni, M., Dinu, G., Kruszewski, G. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In: ACL (1), 238–247 (2014)

De Beule, J. The multiple word guessing game. Belgian J. Linguist 30, (2016)

De Deyne, S., Storms, G. Word associations: Network and semantic properties. Behavior Research Methods, 40 (1), 213–231 Springer (2008)

Harris, Z. S. Distributional structure. Word, 10(2-3), 146–162 (1954)

Heath, D., Norton, D., Ringger, E., Ventura, D. Semantic models as a combination of free association norms and corpus-based correlations. In: Semantic Computing (ICSC), 2013 IEEE Seventh International Conference, 48–55 IEEE (2013)

Mikolov, T., Chen, K., Corrado, G., Dean, J. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781, (2013)

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

Peirsman, Y., Heylen, K., Geeraerts, D. Size matters: tight and loose context definitions in English word space models. In: Proceedings of the ESSLLI workshop on distributional lexical semantics, 34–41 Springer (2008)

Turney, P. D., Pantel, P. From frequency to meaning: Vector space models of semantics. Journal of artificial intelligence research 37, 141–188 (2010)

Von Ahn, L., Kedia, M., Blum, M. Verbosity: a game for collecting common-sense facts. In: Proceedings of the SIGCHI conference on Human Factors in computing systems, 75—78 ACM (2006)