EasyESA: A Low-effort Infrastructure for Explicit Semantic Analysis

Danilo Carvalho[1,2], Çağatay Çallı[3], André Freitas[1], Edward Curry[1]

[1]Insight Centre for Data Analytics, National University of Ireland, Galway
[2]PESC/COPPE, Federal University of Rio de Janeiro (UFRJ)
[3]Department of Computer Engineering, METU, Ankara

1 Introduction

Distributional semantic models (DSMs) are semantic models which are based on the statistical analysis of co-occurrences of words in large corpora. DSMs can be used in a wide spectrum of semantic applications including semantic search, question answering, paraphrase detection, word sense disambiguation, among others. The ability to automatically harvest meaning from unstructured heterogeneous data, its simplicity of use and the ability to build comprehensive semantic models are major strengths of distributional approaches.

The construction of distributional models, however, is dependent on processing large-scale corpora. The English version of Wikipedia 2014, for example, contains 44 GB of article data. The hardware and software infrastructure requirements necessary to process large-scale corpora bring high entry barriers for researchers and developers to start experimenting with distributional semantics. In order to reduce these barriers we developed EasyESA, a high-performance and easy-to-deploy distributional semantics framework and service which provides an Explicit Semantic Analysis (ESA) [4] infrastructure.

2 Explicit Semantic Analysis (ESA)

DSMs are represented as a *vector space model*, where each dimension represents a *context* $\mathcal{C}$ for the linguistic context in which the *target word* occurs in a reference corpus. A *context* can be defined using documents, co-occurrence window sizes (number of neighbouring words or data elements) or syntactic features. The *distributional interpretation* of a target word is defined by a weighted vector of the contexts in which the word occurs, defining a geometric interpretation under a distributional vector space. The weights associated with the vectors are defined using an *associated weighting scheme* $\mathcal{W}$, which can recalibrate the relevance of more generic or discriminative contexts. A *semantic relatedness measure* $\mathcal{S}$ between two words can be calculated by using different *similarity/distance* measures such as the *cosine similarity* or *Euclidean distance*.

In the Explicit Semantic Analysis DSM [4], Wikipedia is used as a reference corpus and the contexts are defined by each Wikipedia article. The weighting scheme is defined by TF/IDF (term frequency/inverse document frequency) and

the similarity measure by the *cosine similarity*. The interpretation vector of a term on ESA is a weighted vector of Wikipedia articles, which is called in the ESA model a concept vector.

A keyword query over the ESA semantic space returns the list of ranked articles titles, which define a concept vector associated with the query terms (where each vector component receives a relevance score). The approach supports the interpretation of small text fragments, where the final context vector is the centroid of the words' concept vectors. The ESA semantic relatedness measure between two terms is calculated by computing the cosine similarity between the concept vectors representing the interpretation of the two terms.

3 EasyESA

EasyESA consists of an open source platform that can be used as a remote service or can be deployed locally. The API consists of three services:
Semantic relatedness measure: Calculates the *semantic relatedness measure* between two terms. The semantic relatedness measure is a real number in the [0,1] interval, representing the degree of semantic proximity between two terms. Semantic relatedness measures are comparative measures and are useful when sets of terms are compared in relation to their semantic proximity. Semantic relatedness can be used for semantic matching in the context of the development of semantic systems such as question answering, text entailment, event matching and semantic search.

- *Example:* Request for the semantic relatedness measure between the words *wife* and *spouse*.
- *Service URL:* http://vmdeb20.deri.ie:8890/esaservice?task=esa&term1=wife&term2=spouse

Concept vector: Given a term, it returns the associated concept vector: a weighted vector of contexts (Wikipedia articles). The term can contain multiple words. The concept vectors can be used to build semantic indexes, which can be applied for semantic applications which depends on high performance semantic matching. An example of a semantic index built using ESA concept vectors is available in [1].

- *Example:* Request for the concept vector of the word *wife* with maximum dimensionality of 50.
- *Service URL:* http://vmdeb20.deri.ie:8890/esaservice?task=vector&source=wife&limit=50

Query explanation: Given two terms, returns the overlap between the concept vectors.

- *Example:* Request for the concept vector overlap between the words *wife* and *spouse* for concept vectors with 100 dimensions.
- *Service URL:* http://vmdeb20.deri.ie:8890/esaservice?task=explain&term1=wife&term2=spouse&limit=100

Mean request values are 0.055 ms for the semantic relatedness measure and 0.080 ms for the concept vector (500 dimensions) on an Intel Core i7 Quad Core 3770 3.40 GHz 32GB DDR3 RAM computer.

EasyESA was developed using Wikiprep-ESA[1] as a basis. The software is available as an open source tool at `http://easy-esa.org`. The improvements targeted the following contributions: (i) major performance improvements (fundamental for the application of distributional semantics in real applications which depends on 100s of requests per second); (ii) robust concurrent queries; (iii) RESTful service API; (iv) deployment of an online service infrastructure; (v) packaging and pre-processed files for easy deployment of a local ESA infrastructure. A detailed description of the improvements can be found at `http://easy-esa.org/improvements`.

4 Demonstrations

Two demonstration applications were built using EasyESA targeting to show the low effort involved in the use of distributional semantics in the context of different semantic tasks. In the demonstration, Wikipedia 2013 was used as a reference corpus. Videos of the running applications are available at: `http://treo.deri.ie/iswc2014demo`

Semantic Search: The first demonstration consists is the use of EasyESA for simulating a semantic search application. In this scenario users can enter a set of terms which can represent the searchable items (for example film genres). Each term associated with a genre has a distributional conceptual representation, i.e. is represented by a concept vector. Users can then enter a search term which has no lexical similarity to the indexed terms. The demonstration computes the semantic relatedness for each vector, ranking the results by their degree of semantic relatedness. In the example, the search query *'winston churchill'* returns the most likely film genres which are associated with the query. Genres *'war'*, *'documentary'*, and *'historical'* were the top most related terms. Figure 1 shows a screenshot of the interface of the example. The demonstration application can be accessed in: `http://vmdeb20.deri.ie/esa-demo/semsearch.html`.

Word Sense Disambiguation (WSD): In the second demonstration, EasyESA is used to perform a word sense disambiguation task (WSD). The user enters a sentence and then selects a word to get the correct sense from WordNet. The WSD application gets the *sentence context* (the words surrounding the target word) finds its associated context vector and computes the semantic relatedness measure in relation to the context vector of the *associated WordNet glosses* for each word sense available. The different senses are then ranked by their semantic relatedness values. In the examples there is no lexical overlap between the sentence context and the different WordNet glosses, with the distributional knowledge from Wikipedia filling the semantic gap between the context and the glosses. The demonstration application can be accessed in: `http://vmdeb20.deri.ie/esa-demo/sensedisambig.html`.

[1] `https://github.com/faraday/wikiprep-esa`

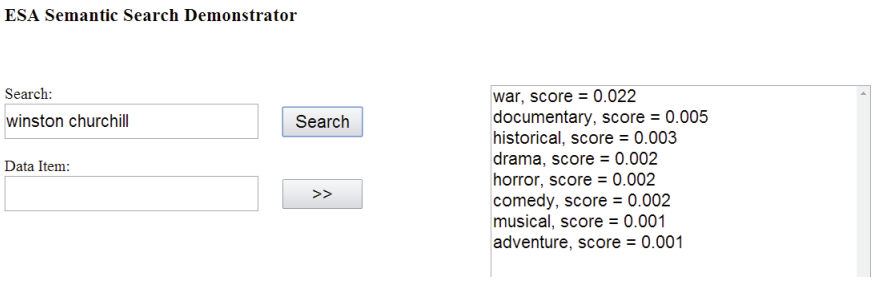

Fig. 1: Screenshot of the semantic search application.

5 Applications using EasyESA

While the demonstration focuses on providing easy to replicate applications for users to start experimenting with distributional semantics, more complex applications were built using EasyESA. Freitas et al [2] used EasyESA for terminology-level search over Linked Data vocabularies, achieving better performance in the semantic matching process when compared to WordNet-based query expansion approach. EasyESA was used in the Treo QA system [1], a schema-agnostic query approach over the Linked Data Web. The system uses a distributional semantics approach to match query terms to dataset elements, supporting schema-agnostic queries. Hasan & Curry [3] use EasyESA for semantically matching complex events from semantically heterogeneous data sources, in a real time scenario.

Acknowledgment: This publication was supported in part by Science Foundation Ireland (SFI) (Grant Number SFI/12/RC/2289) and by the Irish Research Council.

References

1. Freitas, A., Curry, E., Natural Language Queries over Heterogeneous Linked Data Graphs: A Distributional-Compositional Semantics Approach. *In Proc. of the 19th Intl. Conf. on Intelligent User Interfaces.* (2014).
2. Freitas, A., Curry, E., O'Riain, S., A Distributional Approach for Terminological Semantic Search on the Linked Data Web. *In Proc. of the 27th ACM Symposium On Applied Computing (SAC), Semantic Web and Applications (SWA).* (2012).
3. Hasan, S., Curry, E., Approximate Semantic Matching of Events for The Internet of Things,. *In ACM Transactions on Internet Technology (TOIT).* (2014).
4. Gabrilovich, E., Markovitch S., Computing semantic relatedness using Wikipedia-based explicit semantic analysis. *In Proc. of the 20th Intl. Joint Conf. on Artificial Intelligence*, 1606–1611. (2007).