|  |
| --- |
| **Final project literature review** |
| **Self-supervised pipeline for automated product replacement based on neural language models** |
| **Andrei Ionut Damian**  **XCS224U Spring 2020** |
|  |
|  |
| |  | | --- | | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50 | |
|  |

Abstract

|  |
| --- |
| 51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100 |

In past years we have seen a lot of cross-domain application from various deep learning areas to particular real-life cases with apparently little connection. One of those areas is that of applying deep representation learning based on neural language processing to business analytics system. This document presents the summarization and synthesis of several approaches in the area of Natural Language Understanding as well as several papers relevant to the current state of the art in the area of recommender systems based on neural language processing and representation learning in general. This cross-domain analysis is required to further define and test the proposed hypothesis of inferring product replacement retail and distribution systems. We will try to review the proposed natural language understanding papers as tools for our experiments, as well as define a potential outline of the envisioned representation learning pipeline for product replacement inference. The proposed range of analyzed research subjects will be range from word-embedding vector space generations, retrofitting methods for word embedding required to better capture semantic relationship up to analyzing two different views on the problem of product recommender systems.

Introduction and problem definition

# Deep representation learning is arguably one of the most important areas of machine learning as it allows us to automatically – and potentially in un-supervised fashion – discover rich features of the raw data. Since the introduction of the well-known word-embeddings generation algorithms in early 2015, the area on natural language processing and understanding has seen a tremendous improvement. Although it might seem that the actual state-of-the-art resides strongly on contextual embeddings and/or complex multi-head self-attention architectures, it is all based on the initial “basic” steps in the area of semantic vector space models.

# Based on this rich area of research and experimentation a lot of development has been done in the area of applying NLP/NLU approaches to commercial transaction systems including recommender systems. One the most common approaches, further described later in section 2, is to view a retail transaction (a list of purchased items) as a natural language object – a context window, the whole retail transactional database as the actual text corpus and the item SKU database as the “word vocabulary”. This approach of creating a hypothesis similarity between the natural language representation learning and business analytics systems has proved successful in various cases and it is still a very active area of research.

## General problem intuition and list of related papers

# The intuition behind our proposed deep learning pipeline is that we could, at least in theory, generate, through multiple modeling iterations, powerful-enough semantic vector space embeddings for each individual item (product) so that we can infer replacements (synonym) items and propose them in the case on original item shortages – all of these in a self-supervised setting. In order to summarize our proposed papers we have the following lists of review objectives:

|  |
| --- |
| 101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149  150 |

|  |
| --- |
| 151  152  153  154  155  156  157  158  159  160  161  162  163  164  165  166  167  168  169  170  171  172  173  174  175  176  177  178  179  180  181  182  183  184  185  186  187  188  189  190  191  192  193  194  195  196  197  198  199  200 |

1. Semantic vector space model generation approaches such as *GloVe, skip-gram, BOW* (*Word2Vect)*
2. Cross-domain application of NLU in models for product recommendation systems
3. Comparison with state-of-the-art in product recommendation systems based on classic approaches
4. Basic retrofitting techniques
5. Counter-fitting approaches
6. Advanced (functional) retrofitting
7. Other semantic vector space enrichment approaches

# Papers summaries

In the following sub-sections, a deeper look will be given at various approaches and “tools” we plan to employ in our proposed experiment. We will analyze a mix of fundaments of deep representation learning including but not limited to semantic vector space model generation as well as representation learning based recommender systems. As our target is to clearly define our related concepts and approaches, we will present each approach in a general perspective and also pinpoint our take-aways.

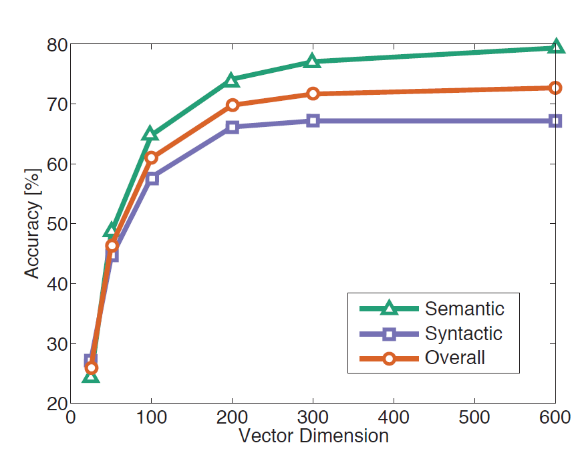
Figure - Accuracy on a specific task as a function of vector size. Image from original paper by Pennington et al

## Semantic vector space models

One of the most important papers relevant to our subject is *“GloVe: Global Vectors for Word Representation”* (Pennington, Socher, & Manning, 2014) as it describes one the core representation learning methods required to generate sound semantic word embeddings beside the similarly well-known *word2vect* (Mikolov, Chen, Corrado, & Dean, 2013). The main intuition behind *GloVe* is based on the fact that statistics of word co-occurrence in a corpus is the primary source of information of the proposed unsupervised method for learning word representations. Thus, the proposed word vector generation pipeline first constructs the matrix of co-occurrence then applies a weighted least squares regression using as target the natural logarithm of the word counts as defined by the below objective function.

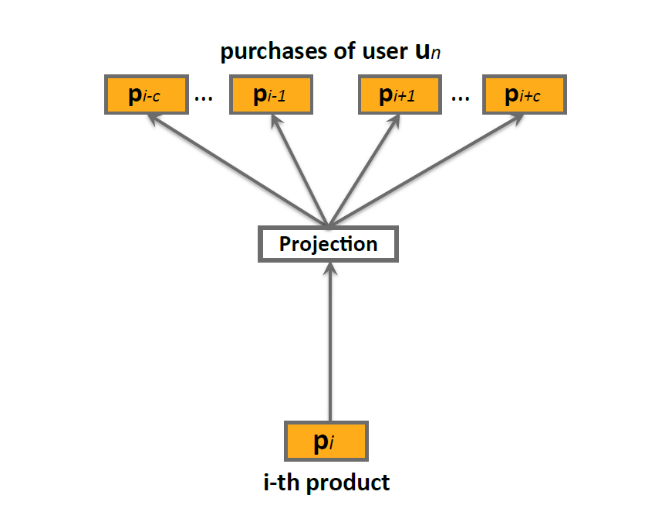
()

Finally, as the model generates through the optimization process to sets of word vectors (*W* and ), the *GloVe* algorithm summarizes the two matrices in order to obtain the final semantic vector space models.

Several other aspects of interest can be found in the paper for the construction of our hypothesis and one of the most important is the correlation between word vector dimension (and thus model capacity) and the task-related accuracy (see Figure 1).

## Product recommendations

The logical next step in our related work review would be to analyze one of the early research papers that proposed the encoding of item SKUs in a similar manner to that of words

Grbovic et al proposes in the paper “*E-commerce in your inbox: Product recommendations at scale*” (Grbovic, et al., 2015) a direct approach of constructing product embeddings based on word2vec skip-gram model.

|  |
| --- |
| 201  202  203  204  205  206  207  208  209  210  211  212  213  214  215  216  217  218  219  220  221  222  223  224  225  226  227  228  229  230  231  232  233  234  235  236  237  238  239  240  241  242  243  244  245  246  247  248  249  250 |

Figure - The skip-gram architecture similar with original word2vec paper. Image from original paper by *Grbovic et al*

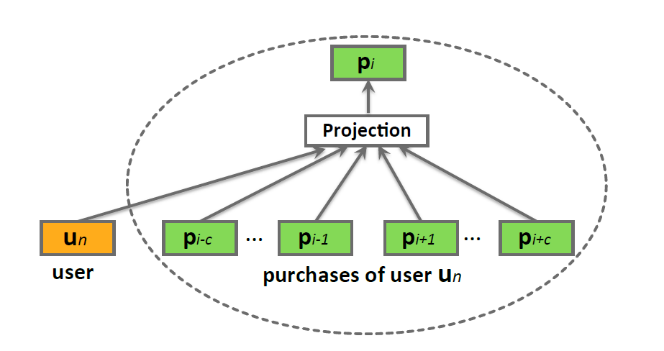
The authors propose a direct analogy between a certain product (item) in a basket and the focal word in the skip-gram context as presented in Figure 2.

Figure - Bag-of-words like approach to enrich the vector space model with user meta information. Picture based on the original paper by Grbovic et al

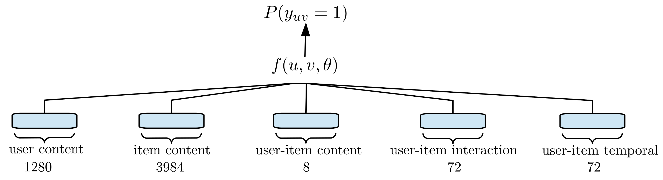
Another proposed aspect was that of enriching the vector space model with meta information such as user-identification in a similar method with *doc2vec* approach – a follow-up of the original word2vec – by Le and Mikolov (Le & Mikolov, 2014). In Figure 3 is presented this particular approach, this time being based on the BOW algorithm (predicting a focal item based on the context) instead of the skip-gram (predicting the context starting from a focal item).

Figure - Figure from paper of Volkovs et all presenting the overall architecture of their classification model using over 4000 hand-crafter features

|  |
| --- |
| 251  252  253  254  255  256  257  258  259  260  261  262  263  264  265  266  267  268  269  270  271  272  273  274  275  276  277  278  279  280  281  282  283  284  285  286  287  288  289  290  291  292  293  294  295  296  297  298  299  300 |

In terms of actual vector space model applications, the authors propose two different hypotheses:

* straight correlation between items cosine distances and the actual real-life similarities of the products (namely the **prod2vec-topK** approach)
* a second more elaborate heuristic approach of finding viable “complementarity” baskets of products that might be sold well together. This second proposed predictive model, called **prod2vec-cluster** by the authors, leverages the hypothesis that similar items can be grouped together using distance metrics with K-Means algorithm. Furthermore, close clusters generated in this manner yield potentially well correlated products and picking items from different close clusters will generate a well-defined complementarity basket.

This particular work is one of the earliest and most well known in the area of applying representation learning and neural language modelling to the problem of retail recommender system. In contrast with this work, the ACM Recommender Systems Challenge’17 was won by a more recent yet more conservative approach by Volkovs et al. In their paper “*Content-based Neighbor Models for Cold Start in Recommender Systems*” (Volkovs, Yu, & Poutanen, 2017) the authors propose a classic approach based on heavy manual featurization of the input data and full supervised approach.

|  |
| --- |
| 301  302  303  304  305  306  307  308  309  310  311  312  313  314  315  316  317  318  319  320  321  322  323  324  325  326  327  328  329  330  331  332  333  334  335  336  337  338  339  340  341  342  343  344  345  346  347  348  349  350 |

## Retrofitting and counter-fitting

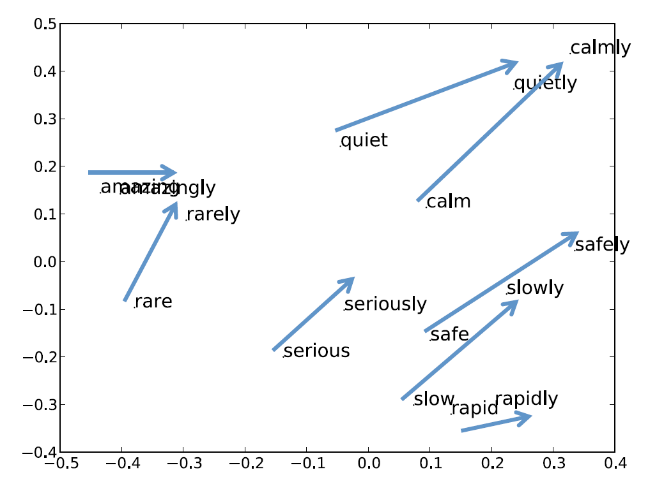
Following the analysis of the base tool – the *GloVe* word-embeddings generation approach and its particular (and potential) cross-domain application to recommender systems - it is now required that we revisit research work in the area of enriching vector space models based on retrofitting and counter-fitting. For this subject we decided to analyze three important papers: “*Retrofitting Word Vectors to Semantic Lexicons*” by Faruqui et al (Faruqui, et al., 2014), “*Counter-fitting word vectors to linguistic constraints*” (Mrkšić, et al., 2016) and the more recent work of Benjamin J. Lengerich, Andrew L. Maas and Christopher Potts “*Retrofitting Distributional Embeddings to Knowledge Graphs with Functional Relations*” (Lengerich, Maas, & Potts, 2017).

Figure - Closing the distance between adjective-adverb pairs. Figure from original paper by Faruqui et al presenting the post-retrofit vector space projection

|  |
| --- |
| 351  352  353  354  355  356  357  358  359  360  361  362  363  364  365  366  367  368  369  370  371  372  373  374  375  376  377  378  379  380  381  382  383  384  385  386  387  388  389  390  391  392  393  394  395  396  397  398  399  400 |

***Basic retrofitting***

In order to better understand the motivation for the review of vector space models retrofitting approaches (and in a later section the review of the counter-fitting approaches) we have to revisit the work done by *Grbovic at al* as well as other teams that pursued the goal of obtaining products semantic vector space models. In all above approaches the authors base their work on the shallow hypothesis that similar items (*words*) measured by cosine distance might have a synonym-like or an entailing-like relationship. However, this as possible for product vector space models as it is with natural language vector space models counterparts. For example, a *GloVe-300* vector space model, pretrained on *Wikipedia* and *Gigaword* might tell us that *'adolescent'*, *'teenager'*, *'puberty'* are very similar to each other based on their pairwise cosine distance and actually ‘*adolescent’* is more similar to ‘*puberty’* than it is to ‘*teenager’*. Exactly the same stands with product semantic vector space models – just using a raw embedding generated with *GloVe* or *word2vect* is not enough to get rich and reliable properties of the items.

This is the point where adding meta-information similar to the *synonymity graphs* in natural language might greatly aid us in further refining the semantic power of our product vector space model. The main intuition is well defined in the work of Faruqui et all as “*graph-based learning technique for using lexical relational resources to obtain higher quality semantic vectors*” by means of post-processing the pre-trained semantic vector space models.

()

Another important aspect is that the proposed “*retrofitting*” optimization method, described by the objective function in equation 2, is agnostic to the origins of the input vector space model, such as original training objective or overall initial training approach. In the this equation *Q* is the new optimized vector space matrix, is the original vector space model, and *i,j*  are pairs of words that are connected by an edge in *E.*

As the authors explain in the paper, they perform experiments using various semantic lexicons such as PPDB (Ganitkevitch, Van Durme, & Callison-Burch, 2013) and WordNet (Miller, 1995), use these in order to improve the word vectors. Following the optimization process they evaluate the quality of the new *retrofitted* word vectors in order to determine how well they capture semantic aspects.

In another paper on the subject of counter-fitting (Mrkšić, et al., 2016) the authors take a more generalized approach by deciding to intuitively pull together the synonymous words while pushing antonymous words vectors apart. The pushing operation proposed by Mrkšić et *al* can be viewed as a similar operation to that proposed by the work of Faruqui etal although the optimization process is slightly different. *The counter-fitting* method proposes a three-part objective function as follows:

|  |
| --- |
| 401  402  403  404  405  406  407  408  409  410  411  412  413  414  415  416  417  418  419  420  421  422  423  424  425  426  427  428  429  430  431  432  433  434  435  436  437  438  439  440  441  442  443  444  445  446  447  448  449  450 |

* the first component, described below in equation (3), of the objective function is responsible of antonym pushing by imposing a minimal *distance* (using a distance function *d*)for the antonym word embeddings (is *max* function)

()

* the second term – equation (4) - purpose is to pull synonym vectors closer by reducing the distance between them to a certain margin and it is closely related with the second term from the objective function in the Faruqui et al work described by equation (2).

()

|  |
| --- |
| 451  452  453  454  455  456  457  458  459  460  461  462  463  464  465  466  467  468  469  470  471  472  473  474  475  476  477  478  479  480  481  482  483  484  485  486  487  488  489  490  491  492  493  494  495  496  497  498  499  500 |

* the third and final component presented in equation (5) has the purpose of preserving the overall vector space structure by minimizing the difference between the original vector space word embedding pairs and the new *counter-fitted* word embeddings. This particular term is quite similar – and has the same purpose - with the first term in the objective function proposed by Faruqui et all (Faruqui, et al., 2014)

()

Finally, the full objective function in equation of the *counter-fitting* approach is simply the weighted sum of the three components as follows:

()

***Short intuitive relationship to our proposed experiment***

Intuitively we plan to employ the same approach of *retrofitting* and *counter-fitting* the *GloVe* generated **product vectors** with existing meta-information in the retail databases – information such as product categories or even detailed category management tree-structures. The hypothesis is that this approach will reduce the cosine distance between products that can actually replace each other in real life in a similar manner as presented in the work of Faruqui et al that addresses word vectors.

***Advanced retrofitting***

Recent work of Benjamin J. Lengerich, Andrew L. Maas and Christopher Potts show that some of the core assumptions of the original base *retrofitting* approach of Faruqui et al, such as that connected entities should have similar embeddings, cannot hold for any real-life applications - such as health knowledge graphs. In order to address the underlined limitations, the authors propose *Functional Retrofitting* - a retrofitting approach that sees pairwise entity relations as functions rather than simple embedding straight similarities. One of the first implication of this approach is that we can include both the *pull* and the *push* between embeddings based on real-life similarity or dissimilarity in a more robust and generalized way that the one presented by *Mrkšić et al*. In order to further understand the approach proposed by the authors of the *Functional Retrofitting* let us understand the objective function in equation (7) by dissecting each of its four components. In order to simplify the explanations, we use a slightly different notation than in the original paper (Lengerich, Maas, & Potts, 2017)

()

()

|  |
| --- |
| 501  502  503  504  505  506  507  508  509  510  511  512  513  514  515  516  517  518  519  520  521  522  523  524  525  526  527  528  529  530  531  532  533  534  535  536  537  538  539  540  541  542  543  544  545  546  547  548  549  550 |

()

()

The first component of the objective function is the semantic vector space model preservation constraint, identical to that in (Faruqui, et al., 2014) and similar to equation (5) of (Mrkšić, et al., 2016). The second and third terms represent the application of relational penalty function for both the positive-relationship knowledge-graph as well as for the *negative space*.

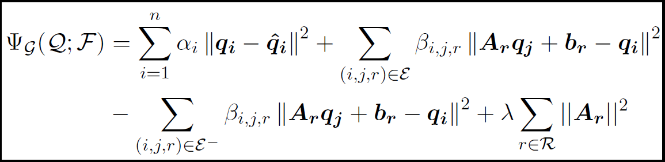
Probably the most important difference between the work of (Faruqui, et al., 2014), (Mrkšić, et al., 2016) and the currently analyzed work of (Lengerich, Maas, & Potts, 2017) if the fact that we can use function-based relationship modelling. The authors of *Functional Retrofitting* propose in their paper two different approaches of the relation modelling – a linear relationship presented in Figure 6 and a basic fully connected neural network with one hidden layer.

Figure - Linear relationship proposed by the authors of *Functional Retrofitting*. Applying *β=0* for *negative space* and using identity as the value for the linear equation coefficient *A* gives us the objective function proposed by (Faruqui, et al., 2014) – image taken from original paper

|  |
| --- |
| 551  552  553  554  555  556  557  558  559  560  561  562  563  564  565  566  567  568  569  570  571  572  573  574  575  576  577  578  579  580  581  582  583  584  585  586  587  588  589  590  591  592  593  594  595  596  597  598  599  600 |

The authors make the source code available at https://github.com/roaminsight/roamresearch.

## Fine-tuning representations generation

The last paper we will analyze in the current literature review for the proposed representation learning *product replacement pipeline* related work is the *Mittens* extension of the *GloVe* algorithm for semantic vector space models generation (Dingwall & Potts, 2018). The main intuition proposed by the authors of this work is that a pre-trained semantic vector space model can be further extended with new vector representations as well as update the existing ones with the new specialized domain data. The authors start from the classic *GloVe* algorithm (Pennington, Socher, & Manning, 2014) and extend it in such a way that a new vector space model is generated from a previous *GloVe* set of embeddings, a new vocabulary and a new matrix of co-occurence. This method preserves the pre-trained *GloVe* vector space model relationships as well as adapts previous embeddings to new relationships and create new embeddings for previously unseen items.

The objective function of the proposed model consists of two parts: the first one being the exact *GloVe* objective function from equation (1) applied to the “new” co-occurrence matrix and the second part being the vector space preservation equation similar to the retrofitting cousins from equations (5) and (8).

# Compare & contrast

Although winner of the ACM Recommender Systems Challenge’17 presented in section 2.2 is entirely based on supervised data and heavy manual featurization, we believe that availability of data as well as relying on well annotated data and supervised datasets will become more and more inefficient for large scale systems deployment with good generalization capacity. Worth to mention is that in the particular approach the model devised by (Volkovs, Yu, & Poutanen, 2017) is not able to map unknown relationships or generate rich semantic knowledge out of unsupervised data.

Regarding *retrofitting* methods we already observed the objective similarity between the work of (Faruqui, et al., 2014) and the (Mrkšić, et al., 2016) where the latter improves upon Faruqui et al objectives with the addition of *antonyms push.*  More interesting seems to be the parallel analysis of the previously mentioned two papers and the work of (Lengerich, Maas, & Potts, 2017). We can view equation (9) in the *Functional Retrofitting* framework as the second term in the objective function from (Faruqui, et al., 2014) and respectively the – equation (4) – from (Mrkšić, et al., 2016). Although equation (10) does not have an intuitive counter-part in equation (2) from (Faruqui, et al., 2014) we can consider that the *anonymity* pushing term, equation (3) from (Mrkšić, et al., 2016), is similar.

|  |
| --- |
| 601  602  603  604  605  606  607  608  609  610  611  612  613  614  615  616  617  618  619  620  621  622  623  624  625  626  627  628  629  630  631  632  633  634  635  636  637  638  639  640  641  642  643  644  645  646  647  648  649  650 |

|  |
| --- |
| 651  652  653  654  655  656  657  658  659  660  661  662  663  664  665  666  667  668  669  670  671  672  673  674  675  676  677  678  679  680  681  682  683  684  685  686  687  688  689  690  691  692  693  694  695  696  697  698  699  700 |

Maybe one of the most important take-aways of this particular paper (Lengerich, Maas, & Potts, 2017) is exactly the cross-domain application – starting from semantic vector space models and knowledge graphs in general and finally using the *Functional Retrofitting* approach to identify potential uses for existing drugs in new diseases where those drugs were not yet employed.

Finally, we can see a clear relationship between the vector space model generation method by (Dingwall & Potts, 2018) and the previously presented *retrofitting* approaches, however this new framework choses to address directly both the problem of retrofitting and new embeddings generation rather than just fine-tuning existing embeddings.

# Future work

For our proposed pipeline for products embeddings, we plan to employ *GloVe* approach to vector space generation as well as use the discounted positive pointwise mutual information taken from the matrix of item co-occurrence. In our further experiments we will try to demonstrate that applying *GloVe* approach will yield similar if not better results than (Grbovic, et al., 2015) in obtaining basic semantic vector space representations of products.

As the intuition suggests from section 2.1 increasing model capacity (bigger vector space dimension) allows capturing richer semantic knowledge that in turns leads to more useful higher-level features and better accuracy down the stream. This is another of the main take-aways for our proposed pipeline – varying and searching for the optimal product representation dimensionality

During our pipeline research and development experiment we plan on applying both direct *retrofitting*-based fine-tuning on the pre-trained product embeddings as well as re-construction of the *GloVe* vector space with new co-occurrence matrices as described by (Dingwall & Potts, 2018).

# 5 References

Dingwall, N., & Potts, C. (2018). *Mittens: An Extension of GloVe for Learning Domain-Specialized.* arXiv preprint arXiv:1803.09901.

Faruqui, M., Dodge, J., Jauhar, S. K., Dyer, C., Hovy, E., & Smith, N. A. (2014). *Retrofitting word vectors to semantic lexicons.* arXiv preprint arXiv:1411.4166.

Ganitkevitch, J., Van Durme, B., & Callison-Burch, C. (2013). PPDB: The paraphrase database. *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, (pp. 758-764).

Grbovic, M., Radosavljevic, V., Djuric, N., Bhamidipati, N., Savla, J., Bhagwan, V., & Sharp, D. (2015). E-commerce in your inbox: Product recommendations at scale. *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 1809-1818.

Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. *International conference on machine learning*, (pp. 1188-1196).

Lengerich, B. J., Maas, A. L., & Potts, C. (2017). *Retrofitting distributional embeddings to knowledge graphs with functional relations.* arXiv preprint arXiv:1708.00112.

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

Miller, G. A. (1995). WordNet: a lexical database for English. *Communications of the ACM*, 39-41.

Mrkšić, N., Séaghdha, D. O., Thomson, B., Gašić, M., Rojas-Barahona, L., Su, P. H., & Young, S. (2016). *Counter-fitting word vectors to linguistic constraints.* arXiv preprint arXiv:1603.00892.

Pennington, J., Socher, R., & Manning, C. (2014). GloVe: Global Vectors forWord Representation. *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, (pp. 1532-1543).

|  |
| --- |
| 701  702  703  704  705  706  707  708  709  710  711  712  713  714  715  716  717  718  719  720  721  722  723  724  725  726  727  728  729  730  731  732  733  734  735  736  737  738  739  740  741  742  743  744  745  746  747  748  749  750 |

Volkovs, M., Yu, G. W., & Poutanen, T. (2017). Content-based Neighbor Models for Cold Start. *In Proceedings of the Recommender Systems Challenge 2017*, (pp. 1-6).