Estimation of the Investability of Real Estate Properties Through Text Analysis

Moloud Shahbazi
Computer Science and Engineering
UC Riverside
mshah008@cs.ucr.edu

Joseph R. Barr
HomeUnion
Irvine, California
joseph@homeunion.com

Vagelis Hristidis
Computer Science and Engineering
UC Riverside
vagelis@cs.ucr.edu

Nani Narayanan Srinivasan
HomeUnion
Irvine, California
nani@homeunion.com

***Abstract*—The Multiple Listing Service, commonly known as the MLS, is the singularly most important database where real estate agents and brokers list real estate properties for sale. It is common that agents include textual comments pertinent to the property. Although the information content of comments varies, it is usually expressed in good faith and in many cases is helpful in shedding light on the overall condition and the value of the property. Therefore, it seems reasonable that semantic text analysis would be useful to evaluate properties, or aspects thereof. As far as we're aware of, no methodology to effectively extract insight from the MLS textual portion exists. In this paper we demonstrate how textual descriptions may be exploited for property ranking. The proposed methodology, which combines supervised and unsupervised methods, identifies domain-specific concepts and combines their contributions to assign a score to a listing. We evaluate the proposed methods using both human evaluators and data-driven evaluation metrics on real datasets (complied from actual listings), and compare them to baseline approaches.**

Keywords*—*text mining, ranking, concept extraction, real estate.

I. Introduction

Real estate agents and other real estate professionals have to sift through hundreds or thousands of listings per day to locate the ones that they should focus on to satisfy their clients, for rent, sale or investment purposes. For example, at HomeUnion [1], expert real estate investment consultants have to carefully read through hundreds of listings per day to pick the most investment-worthy ones. In addition to structured attributes like year-built and number of bedrooms, agents have to read through listing's description, typically a few paragraphs-long as well as related comments from other agents. This is clearly time consuming, translating into significant labor costs.

Evidently, the ranking the listings solely based on the structured attributes is suboptimal as many significant pieces of information are only reflected in the text description; this includes things like remodeling information (new granite counter-top), financial conditions (short sale, foreclosure), etc. Conversely, for one reason or another, an agent will disclose that the home has "foundational issue", clearly a negative home attribute. To our best knowledge, current property valuation methods do not utilize this kind of information for ranking properties.

In this paper, we propose a novel methodology to assign a "goodness" score based on the textual description of a listing. This score can then be combined with other structured attributes to generate an overall goodness score for a property whereby enabling the agent to focus on other, perhaps more pertinent home characteristics.

Specifically, we first build a collection of real-estate-specific concept, a phrase that describes a real-world entity, such as "granite counter top," to use for annotation. Each concept in this collection is manually labeled with a numeric goodness value by experts. For example, "sell as-is" is a negative concept for 'turn-key' investors not wishing to further invest to enhancing the condition of a property (like replacing old carpets.) To that end we've created a continuous vector representations for vocabulary words by analyzing a corpus of real estate descriptions in order to annotate the property's textual information with scored concepts.

Existing real estate lexicons include a limited real estate vocabulary, i.e., concepts, as they are mainly designed for structured data aggregation rather than text analysis [3]. Additionally, these lexicons aren't up-to-date with the current real estate market trends. For example "keyless entry" concept is a newer trend in housing market. As a result it's important to have a method to build a comprehensive concept collection while keeping it up-to-date.

Considering all vocabulary words and n-grams (a phrase of n consecutive words) extracted from a corpus of domain-specific data is an expensive manual labeling task, involves assigning goodness scores to all of them. Instead, we've identified the most useful n-grams by using a measure based on their frequency, their 'chances' to appear together. We then filter the n-grams by removing the non frequent words and phrases with low mutual information. We group similar words and phrases together so that domain experts label similar groups of phrases together as a concept and assign a goodness score to it.

Even if a comprehensive scored lexicon of concepts were available, it will still remain a challenge to identify approximate matches for these concepts, e.g., "kitchen countertop" should match "kitchen counter." In this paper we introduce an unsupervised method for text annotation using word2vec, a neural network modeling framework.

This paper has the following contributions:

- We (intelligently) select a relatively small set of candidate phrases out of a large corpus of real estate text descriptions for domain experts to label as domain concepts, in Section III-A.
- We extract exact and approximate scored real estate concepts from real estate description text and assign a score to the text using word2vec generated word vectors and matching techniques, in Section III-B.
- We evaluate our description scores computed by our algorithms with both human annotators' judgements,

in Section IV-A, and a data-driven approach, in Section IV-C.

Related work is presented in Section II and we conclude in Section V.

II. Related Work

Real Estate Appraisal: There has been research on estimating real estate appraisal, the process of valuing the propertys market value. These studies focus on feature design and price tracking by comparing similar properties or change of price over time. [10] [11] use a learn-to-rank model to predict the ranking (in terms of its potential investment value) of a residential real estate based on features extracted from disparate datasets, such as taxi trajectories, road networks, and online social media ratings. Other studies work on price-rate ratio and price-income ratio for evaluating property values [17]. Some studies rely on financial time series analysis by analysing the trend, periodicity and volatility of house prices [23]. More traditional works are based on repeat sales methods that construct a predefined price index based on properties sold more than once during the given period [29]. The characteristic based methods assume the price of a property merely depends on its characteristics and location [30]. Downie et al. [8] studied the automated valuation models which aggregate and analyze physical characteristics and sales prices of comparable properties to provide property valuations.

More recent works [16], [6] apply general additive mode, support vector machine regression, multilayer perceptron, ranking and clustering ensemble method to computational house valuation. Another study focus on exploiting the mutual enhancement between ranking and clustering to model geographic utility, popularity and influence of latent business area for estimating estate value [11]. They identify and jointly capture the geographical individual, peer, and zone dependencies as an estate-specific ranking objective for enhancing prediction of estate value.

Multiple Listings Service, MLS, is a real estate listing database that provides real estate listings located all across the USA through advertised real estate by real estate agents. It contains real estate MLS listings for rent or sale by Realtors and other realty professionals that are members of local MLS Multiple Listing Service [2]. Currently there are 51 local MLS databases for different USA states. The data for each listing includes a set of structured attributes such as number of bedrooms and square feet that are used as features in real estate valuation techniques. In addition to these attributes, there are two main textual fields that contain property description and real estate agent's remarks about the listed property. These textual data often includes key points regarding the property that affects an agent's judgement while pricing the property. To our knowledge, there has not been any research done regarding extracting these features and analysing them along with other property features. In this project, we employ natural processing and text mining methods to extract key concepts from textual property descriptions and compute an additional numerical feature value describing the property value based on the textual data.

Concept Extraction Methods: Although automatic annotation of online textual resources has been studied extensively in research communities, it still remains a challenging task [6]. Several research studies focus on incorporating natural language processing techniques to do annotation tasks. Most of these studies rely on pre-annotated training examples where they tag sub-strings of the document with pre-defined annotations by identifying the distribution of vocabulary words for different annotation topics [7]. Other studies and tools rely on extra knowledge bases such as regular expression based rules [15] [19] [25] [26] [20] [4]. Another category of studies on text annotation are using machine learning methods to automatically learn the patterns for a text annotations [18]. MnM [31] is a system that retrieves patterns and rules for semantic annotation from a corpus of pre-annotated text. [13] [12] [9] are other examples of automatic rule finding approaches. two main challenges of the existing methods is their dependence on predefined rules or pre-annotated training data as well as assumption of existing concept repository.

Domain specific concepts collection (ontology): Building a collection of labels to use for text annotation has been a challenging task. Constructing ontology is a domain specific task and varies in different domain and contexts. Popescu et al. [24] introduce OPINE, a review-mining system that uses relaxation labeling to find the semantic polarity of words in the context of given product features in online reviews. It first finds features and their attribute and then uses relaxation labeling to extract their polarity in a textual review. Further research is done towards enhancing the ontology quality by refinement to better suit the target domain [27] [28] [5].

III. Concepts Extraction and Description Scoring

In this section we describe the steps we take towards assigning a goodness score to each description text (retrieved from MLS property listings). Our approach consists of two main steps. Figure 1 shows the flow of our approach towards extracting concepts from real estate description text.

First, in Section III-A, we build a collection of real estate-related concepts along with their goodness score. Specifically, after cleaning a corpus of text descriptions of MLS listings and identifying key phrases, the vocabulary is further pruned based on mutual information and frequency. Then, to facilitate the definition of concepts (several phrases may map to the same concept), we cluster the phrases using word2vec vector representations. This allows human labelers to view similar phrases next to each other and mark the ones that should be part of the same concept. In addition, human experts assign a goodness score to each concept.

Next, in Section III-B, given a description text, we extract the scored concepts and compute an aggregate goodness score for the input text. In this phase, a scored concept collection (product of training phase) as well as numerical vector representations are used to annotate a given text with exact or approximate matching of the scored concepts. The aggregation of the concept goodness scores that are extracted from the text generates a single score for the given description.

A. Building Scored Concepts Collection

In this section we show how, given a collection of descriptions, we generate a set of phrases that represent concepts that the real estate experts will score. The goal is to minimize the human effort, while at the same time capturing a large percentage of the concepts mention in property listings.

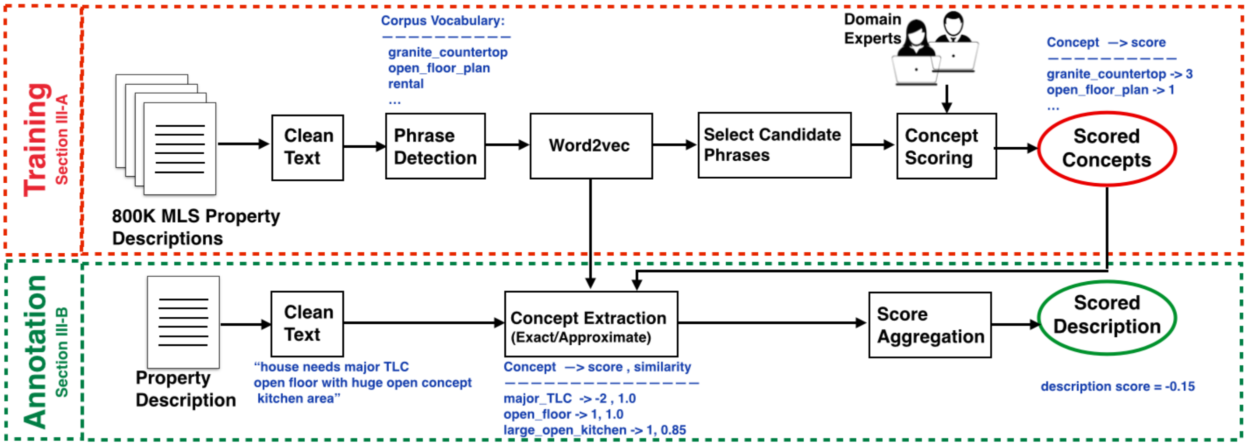


Fig. 1 Text annotation architecture. Example texts in the diagram are colored blue.

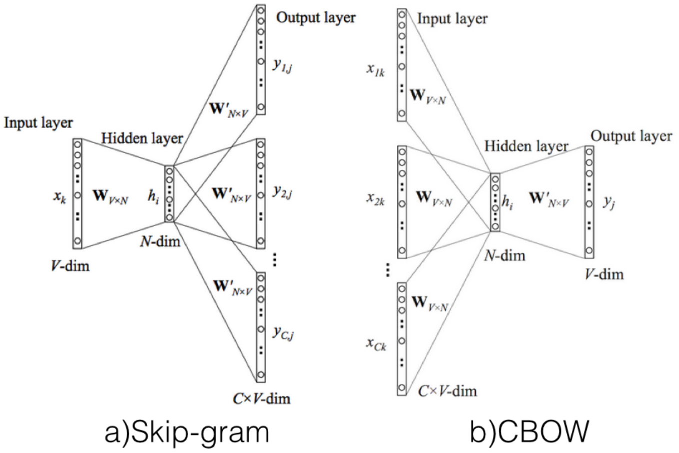
We use a corpus of 800K property listing descriptions and remarks retrieved from MLS database. we clean the text in corpus by lemmatization, lower-casing, tokenizing and removing the stop words (a set of 30 words that we collected) for each property. Next, we identify the phrases by merging the words that have a frequently happen together and infrequently happen in other context with a simple data-driven approach [21]. To be precise, for each consecutive pair of words (bigram) we compute a score using Equation 1 defined as follows:

$$score(a\,b) = \frac{(p_{ab} - min\_count)}{p_a \times p_b} \tag{1}$$

where p_a, p_b are frequency of terms "a", and "b" respectively, and p_{ab} is the frequency of phrase "a b" in the corpus. min_count is a parameter to account for minimum term frequency. If a pair has a score that is greater than a threshold score (a parameter for phrase generation) then words will be attached using a hyphen and a phrase in the corpus is created. Algorithm 1 shows how word2phrase works. We run word2phrase three times in order to with decreasing min_count threshold to allow longer phrases consisting multiple words. The outputs of fist and second runs are the inputs for second and third runs respectively. Figure 2 shows an example of running word2phrase twice. In first round "steel" and "appliances" meet phrase score threshold and get attached as a phrase. In second round, "stainless" and "steel-appliances" are the two words that are grouped together create a phrase.

Round 1: Kitchen has stainless steel_appliances.
Round 2: Kitchen has stainless_steel_appliances.

Fig. 2 Example of word to phrase transformation. Round 1 identifies ”steel” and ”appliances” to be connected as phrase. In Round 2, ”steel_appliances” is considered as one word and it is identified as a phrase in combination with ”stainless”.



a)Skip-gram b)CBOW

Fig. 3 Two word2vec learning approaches. a) skip-gram model learns a model that given a word, guesses the context. b) CBOW (continuous bag of words) model learns to guess the word given a context. In these diagrams, V is the number of vocabulary words and N is the size of word vectors.

After identifying phrases and merging them into one unit using word2phrase method, we use word2vec [22] tool to learn the space of continuous word and phrase representations from the preprocessed corpus. word2vec provides an efficient implementation of the continuous bag-of-words and skip-gram architectures for computing vector representations of words. Word2vec allows us to train models on a large data sets (up to hundreds of billions of words). Word2vec computes a vector representation for each word using a recurrent neural network. Figure 3 shows two main approaches that word2vec uses for training the model to learn word representations. The training objective of the Skip-gram model is to find word representations that are useful for predicting the surrounding

Algorithm 1 word2phrase algorithm. T is the score threshold for generating a phrase by attaching two words.

```
procedure WORD2PHRASE($text$)
    text = clean(text)
    List tokens = tokenize(text)
    for $(i = 1; i <= tokens.size(); i++)$ do
        if $Score(tokens[i-1], tokens[i]) > T$ then
            new-phrase $= tokens[i-1] + “\_” + tokens[i]$
        end if
    end for
end procedure
```

words in a text window. While in CBOW (continuous bag of words), the model is trained such that it predicts a word given its context (surrounding words).

The input of the skip-gram model is a single word w_I and the output is the words in w_I's context $\{w_{O,1}, ..., w_{O,C}\}$ defined by a word window of size C. For example, consider the sentence "I drove my car to the store". A potential training instance could be the word "walking-distance" as an input and the words "curb-appeal","propery","located","school","church","shopping-mall" as outputs. All of these words are one-hot encoded, meaning they are vectors of length V (the size of the vocabulary) with a value of 1 at the index corresponding to the word and zeros in all other indexes. As we can see, Word2vec is essentially creating training examples from plain text which means that we can have a virtually unlimited number of training examples at our disposal. In CBOW version, the input layer consists of the one-hot encoded input context words $\{x_1, ..., x_C\}$ for a word window of size C and vocabulary of size V. the output layer is output word y in the training example which is also one-hot encoded. The word vector representations are the weights of the neural network and they are learned after the training cycle is complete.

After training high dimensional word vectors on a large amount of data, the resulting vectors can be used to answer very subtle semantic relationships between words [22]. More specifically, the words that are semantically related such as synonyms or the words of the same category tend to have very similar vectors because they appear in the same context. Cosine similarity measure is used to quantify the similarity of the words based on their vectors.

We use Skip-gram model in training phase to compute word vector representations. Using cosine similarity measure, we compute word clusters with KNN method to group the relevant words together. The goal of clustering is to identify groups of relevant concepts and make the labeling task easier for the domain experts by putting all the relevant concepts together. We further prune these clusters by removing non frequent words and phrases with low phrase scores (ex. mutual information or Equation 1) to make a smaller set for the domain experts to label the goodness score.In this project, after processing 800K property descriptions, we come up with approximately 3000 candidate concepts grouped in 500 clusters to be scored by domain experts.

Labeling the goodness of concepts: We asked two real estate experts to merge together phrases with the same meaning, in the context of homes evaluation, to form concepts, that is, "kitchen counter" and "kitchen countertop." Further, we asked them assign a goodness score between -10 and 10 to each of the 3000 concepts. For example, since "foundation_issue" is costly for the property, it get a -8 while "mior_cosmetics" get -1 as their goodness score. The average score of multiple expert's opinion was recorded for each concept.

B. Scoring the Property Listings

Now that we have the scored collection of concepts, in order to score a property's textual description, we need to detect the real estate concepts in the text and compute the overall score based on their aggregated goodness scores. In the previous section, we explained how we build a collection of concepts that we use to annotate the text. An exact concept may not be exactly present in the text but a semantically similar term may be present (e.g., synonyms, acronyms, etc.). In this case, the word2vec word vectors can be used to capture the relevance of the words to the real estate concepts in our concepts collection. In this section, we describe three methods to extract the concepts from a given text.

The goal is to extract labeled concepts and assign a score to the property description based on the aggregation of the concepts' goodness scores. We propose three variations of our solution with different trade-off's between computation time and precision.

1) Exact concept matching (ECM): Given a property description, we find all the scored concepts that exactly appear in the cleaned text. In this variation, we define the property score as the summation of the goodness score of the found concepts:

$$score(text) = \sum_{concept \in text} goodness(concept)$$

2) Nearest Concepts Matching (NCM): In this variation, we find all the vocabulary words/phrases that exist in the input text string. Then, for each word/phrase, we find the nearest concept with the maximum cosine similarity amongst the scored concepts. If the cosine similarity is greater than a threshold parameter, then the concept will be considered as found with a weight equal to its similarity to the existing word/phrase. Similarity threshold is a parameter in our system. The textual score of the property is computed by summing up the goodness scores of the found concepts weighted by their similarity to the vocabulary word/phrase exist in the text (in terms of cosine similarity). Formally, we define the score of textual property description as follows:

$$score(text) = \\ \sum_{p \in text} goodness(nearest\_concept(p)) \times weight(p)$$

where p is a vocabulary word/phrase that is in the text and weight of the word/phrase is defined as follows:

$$weight(p) = cosine\_similarity(nearest\_concept(p), p)$$

3) Non Redundant Nearest Concepts Matching (NRNCM): In NRNCM, we first extract the concepts using nearest concept matching. Then, in a greedy way, we iterate over the set of matched concepts and compare each concept with the rest of concepts. If two concepts of the matched set have a similarity greater than the similarity threshold then we remove the concept with the smaller similarity weight from the set of concepts and continue. The purpose of this pruning is to avoid scoring the property multiple time for the same concept.

IV. Evaluation Results

A. Experimental Setup and Measures

Data Set: Our corpus includes 800K property descriptions and agent remarks that are inserted by real estate agents for property listings (fetched from MLS database). These descriptions are retrieved from the property listings fetched from the MLS database. After pre-processing the corpus to clean the text we get a vocabulary of size 45K and 34485K words in total.

TABLE I Example of ranking of 5 property listing text with human annotator (ground truth) and automatic machine annotator. Optimal ranking in last column is based on assumption that all ranks are distinct and they are in perfect correlation with human ranking. In case of tie in scores, an equal ranking is assigned to all of them.

Human Score	Human Ranking	Machine Score	Machine Ranking	Machine Optimal Ranking
+1	2	23.0	3	1
+1	2	45.3	2	2
+1	3	51.0	1	3
0	3	3.0	4	4
-1	4	-10.0	5	5

Experimental Setup: We set min_count from Equation 1 to be equal to 100. For the phrase score threshold, we chose different values during our experiments and show how the results changes. Finally, We evaluate our scoring algorithms using both CBOW and skip-gram models to train word vectors.

B. Evaluation Using Human Labeled Text

We randomly selected a set of 100 property listings. The real estate experts assign a score of -1, 0 or +1 to each description if it is a negative, neutral or positive description. We then sort the descriptions with scores that are computed using three variations of our proposed solution.

We measure the correlation of the rankings using normalized Kendall-Tau [14] measure. Kendal_Tau measures the ranking agreement of two different ranking schemes by comparing each pair of rankings for a set of ranked objects. Assume that o_1 and o_2 are two objects and their ranks by ranking method R^1 has $R^1_{o_1} < R^1_{o_2}$ relation meaning that o_2 is ranked higher than o_1. If the ranks by R^2 agrees with R^1, such that $R^2_{o_1} < R^2_{o_2}$, then o_1 and o_2 are concordant pairs. On the other hand, if $R^2_{o_1} > R^2_{o_2}$, o_1 and o_2 are a discordant pair. based on this definition, Kendall_Tau is defined as following.

$$Kendall\_Tau = \frac{n_c - n_d}{\sqrt{(n_0 - n_1)(n_0 - n_2)}}$$

where n_c and n_d are the number of concordant and discordant pairs respectively. n_0 is the number of possible pairs. n_1 and n_2 are number of pairs with a tie using ranking methods R^1 and R^2 respectively.

We normalize rank correlation measures by their value for perfect ranking and we call it optimal machine ranking. We assume in optimal machine ranking all the objects have distinct rank and are ordered such that the ranking completely agrees with human ranking. Table I shows an example of ranking using human judgement and our machine score based ranking. Based on these definitions, figures 5 and 4 show the Kendall-tau rank correlation of three proposed algorithms based on CBOW and Skip-gram strategies for training word vectors respectively. We compare our algorithms score based rankings to domain expert judgements. We evaluate the rank correlation for different cosine similarity thresholds between 0.5 and 0.9 inclusive. As shown in figures 5 and 4, the ECM algorithm has a constant correlation for different similarity thresholds since it is independent of this parameter. The correlation of NCM is similar or better than NRCM for all similarity thresholds. This observation implies that redundant mentioning of similar concepts should not be ignored. Another observation is that maximum correlation for NCM and NRNCM is happening in lower threshold while using CBOW based word2vec learning comparing to Skip-gram based learning. The peak correlation also is slighty lower using CBOW. The reason is that using CBOW, similar words' vectors have higher distance than Skip-gram which causes more concepts to be filtered using higher thresholds.

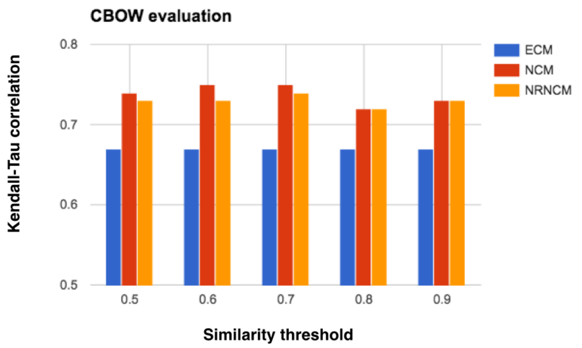


Fig. 4 Kendall_Tau evaluation of ranking algorithms for different cosine similarity thresholds. Word vectors are trained using CBOW model.

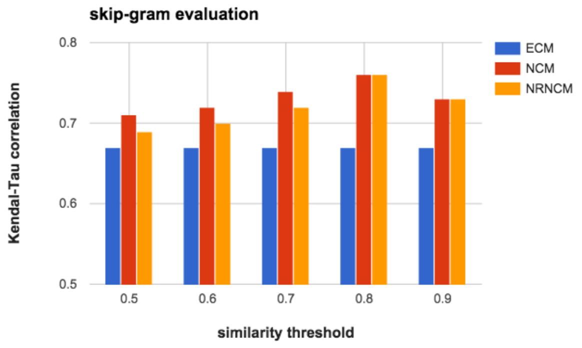


Fig. 5 Kendall_Tau evaluation of ranking algorithms for different cosine similarity thresholds. Word vectors are trained using Skip-gram model.

C. Evaluation Using Price Variation

We did a more extensive evaluation by counting the properties that are expected to have similar listing price but they don't. We assume the reason should be explained in the description text and agent remarks. If two properties are similar in terms of key features including location, year built, lot-size and square feet, we expect them to have the same listing price. For location similarity, we consider properties with similar zipcode and community subdivision. In terms of the square feet and lot size, we consider them similar if their difference is less than 10% of their average.

We use 5000 property listings and find the pairs that are similar for all the property features that we mentioned. for each property we compute a score based on the property description and agent remarks. We found 310 similar pairs out of our sample set.

For each pair, if one of the properties' price is greater than the other property's price with a difference greater than 10%

TABLE II Data-driven ranking evaluation.

	# of properties	# of similar pairs	# price/score agreement
ECM	5000	310	42
NCM	5000	310	73
NRNCM	5000	310	70

of their price average and its text score is also greater, we say that scoring function and listing price have agreement.

Table II shows the results of the evaluation for different proposed algorithms. For NCM and NRCM, Skip-gram based word vectors are used to match approximate concepts with a similarity threshold equal to 0.8. For each algorithm the number of pairs with agreement is listed in the table. This evaluation also shows that performance of NCM is slightly better than NRCM as shown in human judgement base evaluation.

V. Conclusion

In this paper we propose a semi-supervised method for building a comprehensive real estate concept collection using a corpus of property descriptions from MLS. We propose an effective unsupervised method to annotate property descriptions and extract real estate concepts. We use the extracted exact and approximate concepts goodness scores to compute a score for property description. The calculated score is used to rank the property descriptions. We use both human judgements and a data-driven approach to evaluate our algorithms. Our results indicate that the ranking by NCM algorithm (weighted aggregation of exact and approximate concepts) is the most effective method, which has the highest rank correlation of 0.76 with human judgements.

References

[1] HomeUnion Inc. wensite. http://www.homeunion.com.

[2] Multiple Listing Services. http://www.mls.com.

[3] Real Estate Standards Organization. http://www.reso.org.

[4] J. Baumeister, J. Reutelshoefer, and F. Puppe. Knowwe: a semantic wiki for knowledge engineering. *Applied Intelligence*, 35(3):323–344, 2011.

[5] R. Bendaoud, Y. Toussaint, and A. Napoli. Pactole: A methodology and a system for semi-automatically enriching an ontology from a collection of texts. In *Conceptual Structures: Knowledge Visualization and Reasoning*, pages 203–216. Springer, 2008.

[6] J. Cigarrán-Recuero, J. Gayoso-Cabada, M. Rodríguez-Artacho, M.-D. Romero-López, A. Sarasa-Cabezuelo, and J.-L. Sierra. Assessing semantic annotation activities with formal concept analysis. *Expert Systems with Applications*, 41(11):5495–5508, 2014.

[7] S. Dill, N. Eiron, D. Gibson, D. Gruhl, R. Guha, A. Jhingran, T. Kanungo, K. S. McCurley, S. Rajagopalan, A. Tomkins, et al. A case for automated large-scale semantic annotation. *Web Semantics: Science, Services and Agents on the World Wide Web*, 1(1):115–132, 2003.

[8] M.-L. Downie and G. Robson. Automated valuation models: an international perspective. 2008.

[9] O. Etzioni, M. Cafarella, D. Downey, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. Unsupervised named-entity extraction from the web: An experimental study. *Artificial intelligence*, 165(1):91–134, 2005.

[10] Y. Fu, Y. Ge, Y. Zheng, Z. Yao, Y. Liu, H. Xiong, and N. J. Yuan. Sparse real estate ranking with online user reviews and offline moving behaviors. In *Data Mining (ICDM), 2014 IEEE International Conference on*, pages 120–129. IEEE, 2014.

[11] Y. Fu, H. Xiong, Y. Ge, Z. Yao, Y. Zheng, and Z.-H. Zhou. Exploiting geographic dependencies for real estate appraisal: A mutual perspective of ranking and clustering. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1047–1056. ACM, 2014.

[12] G. Giannopoulos, N. Bikakis, T. Dalamagas, and T. Sellis. Gontogle: a tool for semantic annotation and search. In *The Semantic Web: Research and Applications*, pages 376–380. Springer, 2010.

[13] S. Handschuh and S. Staab. Authoring and annotation of web pages in cream. In *Proceedings of the 11th international conference on World Wide Web*, pages 462–473. ACM, 2002.

[14] M. G. Kendall. A new measure of rank correlation. *Biometrika*, pages 81–93, 1938.

[15] P. A. Kogut and W. S. Holmes III. Aerodaml: Applying information extraction to generate daml annotations from web pages. In *Semannot@ K-CAP 2001*, 2001.

[16] V. Kontrimas and A. Verikas. The mass appraisal of the real estate by computational intelligence. *Applied Soft Computing*, 11(1):443–448, 2011.

[17] J. Krainer and C. Wei. House prices and fundamental value. *FRBSF Economic Letter*, 2004.

[18] N. Kushmerick. Wrapper induction: Efficiency and expressiveness. *Artificial Intelligence*, 118(1):15–68, 2000.

[19] S. K. Malik, N. Prakash, and S. Rizvi. Semantic annotation framework for intelligent information retrieval using kim architecture. *International Journal of Web & Semantic Technology (IJWest)*, 1(4):12–26, 2010.

[20] D. Maynard. Multi-source and multilingual information extraction. *Expert Update*, 6(3):11–16, 2003.

[21] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[22] T. Mikolov, W.-t. Yih, and G. Zweig. Linguistic regularities in continuous space word representations. In *HLT-NAACL*, pages 746–751, 2013.

[23] C. H. Nagaraja, L. D. Brown, and L. H. Zhao. An autoregressive approach to house price modeling. *The Annals of Applied Statistics*, pages 124–149, 2011.

[24] A.-M. Popescu and O. Etzioni. Extracting product features and opinions from reviews. In *Natural language processing and text mining*, pages 9–28. Springer, 2007.

[25] B. Popov, A. Kiryakov, A. Kirilov, D. Manov, D. Ognyanoff, and M. Goranov. Kim–semantic annotation platform. In *The Semantic Web-ISWC 2003*, pages 834–849. Springer, 2003.

[26] B. Popov, A. Kiryakov, D. Ognyanoff, D. Manov, and A. Kirilov. Kim-a semantic platform for information extraction and retrieval. *Natural language engineering*, 10(3-4):375–392, 2004.

[27] S. Rudolph, J. Völker, and P. Hitzler. Supporting lexical ontology learning by relational exploration. In *Conceptual Structures: Knowledge Architectures for Smart Applications*, pages 488–491. Springer, 2007.

[28] B. Sertkaya. Ontocomp: A protege plugin for completing owl ontologies. In *The Semantic Web: Research and Applications*, pages 898–902. Springer, 2009.

[29] R. J. Shiller. Arithmetic repeat sales price estimators. *Journal of Housing Economics*, 1(1):110–126, 1991.

[30] L. O. Taylor. The hedonic method. In *A primer on nonmarket valuation*, pages 331–393. Springer, 2003.

[31] M. Vargas-Vera, E. Motta, J. Domingue, M. Lanzoni, A. Stutt, and F. Ciravegna. Mnm: Ontology driven semi-automatic and automatic support for semantic markup. In *Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web*, pages 379–391. Springer, 2002.