

A Review-based Context-Aware Recommender Systems: Using Custom NER and Factorization Machines

Rabie Madani, Abderrahmane Ez-zahout
Intelligent Processing and Security of Systems Team
Computer Sciences Department
Faculty of Sciences
Mohammed V University
Rabat, Morocco

Abstract—Recommender Systems depend fundamentally on user feedback to provide recommendation. Classical Recommenders are based only on historical data and also suffer from several problems linked to the lack of data such as sparsity. Users' reviews represent a massive amount of valuable and rich knowledge information, but they are still ignored by most of current recommender systems. Information such as users' preferences and contextual data could be extracted from reviews and integrated into Recommender Systems to provide more accurate recommendations. In this paper, we present a Context Aware Recommender System model, based on a Bidirectional Encoder Representations from Transformers (BERT) pretrained model to customize Named Entity Recognition (NER). The model allows to automatically extract contextual information from reviews then insert extracted data into a Contextual Machine Factorization to compute and predict ratings. Empirical results show that our model improves the quality of recommendation and outperforms existing Recommender Systems.

Keywords—Recommender systems; context aware recommender systems; factorization machines; bidirectional encoder representations from transformers; named entity recognition

I. INTRODUCTION

In the last few years, the number of services and items offered and produced by businesses and websites have increased quickly, which makes the choice of products and services meeting customers' needs more difficult.

Recommender systems (RS) tackle this problem by helping users to find suitable resources, based on their past behaviors and preferences. Today, companies use RS in several domains to assist users, enhance customer experience and make it easier for them to satisfy their needs by speeding up searches.

Whereas, traditional Recommender systems such as Collaborative Filtering techniques [1] are based on two dimensions (User X Item) and on numeric rating (e.g., 5 stars rating) to compute similarities between users and items and produce recommendations. Context Aware Recommender Systems [2] use other dimensions beside the two classical dimensions, namely contextual information dimensions (User X Item X Context) to enhance the accuracy. The contextual information represents the environmental factors that influence the user's decision. In general, numeric rating expresses whether a user likes or dislikes an item, however, it does not allow us to

understand why, when or where he/she makes this choice and reasons behind it.

Where sparsity and the lack of information represent big challenges to Recommender Systems, customers' reviews could be a good resource to solve these problems and help companies to well understand decisions made by users. In fact, many models have been proposed to extract valuable information from reviews like sentiment analysis and rating extraction, but only few works have been presented to extract contextual information from users' reviews.

In this article, we present a new model for contextual information extraction based on a pre-training language representation method trained on large amounts of data like Wikipedia called BERT and a custom Named Entity Recognition. The extracted data is used by a Contextual Machine Factorization algorithm to predict the user's interest.

This paper is organized as follows.

The remainder of this paper is presented as follows. In Section 2, we give a review of related works. In Section 3, we define the context dimension and we present the Named Entity Recognition and BERT. We introduce Factorization Machine algorithm and its use in Context-Aware Recommender Systems in Section 4. In Section 5, we present the proposed work in detail. In Section 6 we discuss obtained results. Finally, a conclusion of the work is presented in the last section.

II. RELATED WORK

Recently, many works have been proposed for extracting precious data from reviews and integrating them into recommendation process. This section presents recent applications of reviews-based Recommender systems.

Zheng et al. [3] presented a Deep Cooperative Neural Networks (DeepCoNN) model based on word embedding techniques and two convolutional neural networks (CNNs). The first network extracts user behaviors from users' reviews and the second network extracts item properties from reviews written on items. The model merges network outputs and transmits it to a factorization machine algorithm for the prediction.

Similarly to DeepCoNN [3], R. Catherine and W. Cohen [4] proposed a model called Learning to Transform for

Recommendation (TransNets) based on two parallel CNNs, one to process the target review and the other to process the texts of the user and item pair and Factorization Machine to predict rating. The difference between the two models is that the transnet model integrates an additional Transform layer to represent the target user-target item pair.

McAuley and Leskovec [5], introduced a Hidden Factors and Hidden Topics (HFT) model that merges reviews written by users and ratings to provide recommendations. The model uses Latent-Factor Recommender Systems to predict ratings and the Latent Dirichlet Allocation to discover hidden dimensions in review text.

Tan et al. [6], introduced a Rating-Boosted Latent Topics (RBLT) framework which models item features and user preferences by combining textual information extracted from reviews and numeric ratings. The RBLT model represents users item as a latent rating factor distribution, and repeats reviews with rating n time to dominate topics. To perform predictions outputs are introduced into a Latent Factorization Machine (LFM).

Zhang et al. [7], proposed an Explicit Factor Models (EFM) to produce explainable recommendations. EFM extracts user sentiments from reviews and explicit item features, then recommends or not recommends items based on hidden features learned, items features and users' interest. All previously cited works have exploited reviews to boost recommender systems, but they ignore contextual information which could significantly improve recommendations.

Other researchers succeeded in integrating context in recommendation tasks such as Aciar, [8] proposed a Mining Context Information method based on classification rules text mining techniques to automatically identify user's preferences and contextual information inside reviews, extract it and integrate it in recommendation. However, this method identifies sentences containing context but it can not extract contextual information from these sentences.

Hariri et al. [9] which proposed a Context Aware Recommender system that models user reviews to obtain contextual data and combines it with rating history to compute the utility function and suggest items to users. The model handles the context like a supervised problem of topic modeling and builds the classifier of context using a labeled-LDA. The system uses conventional recommendation algorithms to predict ratings. However, this work predicts the utility function not the rating.

Levi et al. [10] introduced a Cold Start Context-Based Hotel Recommender System based on context groups extracted from reviews. This approach uses many elements, including a weighted algorithm for text mining, an analysis to understand hotel features sentiment, clustering to build a hotel's vocabulary and nationality groups. Despite this study tackling the cold start issue, it doesn't show how to integrate extracted context in recommendation adequately.

Compos et al. [11] introduced an approach to extract contextual information from user reviews using large-scale and generic context taxonomy based on semantic entities obtained from DBpedia. In this approach a software tool builds the taxonomy by exploring DBpedia automatically, and also allows for manual adjustments of the taxonomy. Despite this work

presenting a semi-automatic method to extract context from reviews, it does not explain how to use extracted data to predict ratings.

Lahlou et al. [12] proposed a review aware Recommender system based on users' reviews to build a contextual recommendation. The proposed architecture allows to automatically exploit contextual information from reviews to build recommendations. They also presented a Textual Context Aware Factorization Machines (TCAFM) which is tailored to context. This work shows good performances in terms of accuracy, but it considers the whole review as a context instead of extracting contextual data, and in the real world datasets only few reviews contain this kind of data.

III. CONTEXT EXTRACTION

To extract contextual information from reviews, we should firstly define context dimensions (a.k.a. categories of context). In the literature, many context modeling approaches have been introduced, but the most commonly used context representation is [13], the major of these approaches use ontologies to build context taxonomy. For instance, Castelli et al. [14] use the W4 model (a.k.a. Who, When, Where, What) as components of context, "Who" is linked to the Person, "When" is associated to the Time, "Where" refers to the Location and "What" refers to the Fact. Similarly, Kim et al. [15] instantiate the 5W1H model (a.k.a. Who, Why, Where, What, When, How) as contextual components associated respectively to Status, Goal, Location, Role, Time, Action. Chaari et al. [16] proposed a basic context descriptor to describe contextual components as Service, User, Activity, Location, Device, Resource, Network. Table I resumes some principle modeling techniques of context. After revising and analysing proposed works, we choose to use four contextual dimensions, Time, Location, Companion and Environmental dimensions.

TABLE I. MAIN CONTEXT MODELING APPROACHES

Ref	Context categories
[14]	The W4 model (Who, When, Where, What): Person, Time, Location, Fact.
[15]	The 5W1H model (Who, Why, Where, What, When, How): Status, Goal, Location, Role, Time, Action.
[16]	Basic context descriptor: Service, User, Activity, Location, Device, Resource, Network.
[17]	Agent, Action, Time, Space, Policy, Event, Person, Geo-spatial
[11]	Location, Time, Environmental context, Social context

A. Named Entity Recognition (NER)

NER is one of the most important tasks of information extraction in Natural Language Processing. The aim of NER is to identify named entities in unstructured text then classify them into predefined categories (locations, organisations, monetary values, percentages, time, persons, etc.) The Named Entity Recognition is considered as a sequence labeling task

where a given sentence is presented as a tokens sequence $w = (w_1, w_2, w_3, \dots, w_n)$, and transformed to a token labels sequence $y = (y_1, y_2, y_3, \dots, y_n)$ [18], the neural model is generally composed from three elements : word embedding layer, context encoder layer and decoder layer [19].

Bidirectional long short-term memory networks (Bi-LSTM) [20][21] is widely applied in Natural Language Processing tasks and adopted by most of NER, due to its sequential characteristic and its capacity to learn contextual word representations. Despite NER having been employed in several application domains, many application fields are still not discovered, such as in Context Aware Recommender Systems.

B. BERT

BERT (Bidirectional Encoder Representations from Transformers) As its name indicates, it is a model of language representation that relies on a module called "Transformer". A transformer is a component which relies on attention methods and which is built on the basis of an encoder and a decoder. In opposition with directional and shallow-bidirectional models (OpenAI GPT [22], ELMo [23]), BERT pre-trains deep bidirectional representations from unstructured text on both left and right context in all layers [24]. It has been pre-trained on large corpus such as the entire BookCorpus and Wikipedia.

Fig. 1 represents BERT architecture, The arrows show the flow of information from layer to layer. E_1, T_r, m and T_1 respectively represent the embedding representation, the intermediate representations and the final output for a given word.

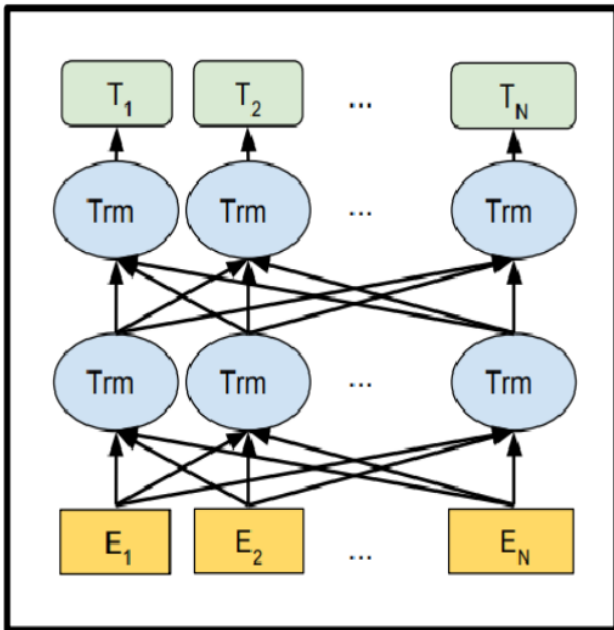


Fig. 1. BERT Architecture.

Language Modeling is a usual NLP task of predicting the next word given the start of the sentence. The Masked Language Model (MLM) allows to BERT to learn in an unsupervised way, the entry is sufficient on its own, no need to

label anything. The principle of Masked Language Modeling is to predict "masked" tokens from the other tokens in the sequence. In the first step of BERT's pre-training, 15% of the tokens of each sequence are masked, randomly. This step is very essential because BERT gets its deep bidirectionality from it.

IV. MACHINE FACTORIZATION FOR CONTEXT AWARE RECOMMENDER SYSTEMS

Factorization machines (FM) proposed by Rendle [25], a general-purpose supervised learning algorithm that could be used in regression and classification tasks. It rapidly became one of the most popular algorithms for recommendation and prediction. It is a generalization of the linear model that is able to capture interactions between variables and also it can significantly reduce the polynomial complexity to linear computation time. FM is very efficient especially within high dimensional sparse datasets.

Let $x \in \mathbf{R}^d$ be the feature vectors and y be the corresponding label. The model equation for a factorization machine is defined as:

$$y(x) = w_0 + \sum_{i=1}^d w_i x_i + \sum_{i=1}^d \sum_{j=i+1}^d \langle v_i, v_j \rangle x_i x_j \quad (1)$$

Where $w_0 \in \mathbf{R}$ is the bias term, $w \in \mathbf{R}^d$ are weights corresponding to each feature vector, $V \in \mathbf{R}^{d \times k}$ the interaction matrix, v_i is the i^{th} row of the V matrix, $\langle v_i, v_j \rangle$ the interaction between the i^{th} and j^{th} variable. It is important to point out that this factorization has the ability to compute all pairwise interactions, even hidden feature interactions which can significantly reduce engineering efforts.

Context Aware Factorization Machines is an application of the origin FM algorithm without any tuning. In effect, it is easy for the algorithm to incorporate the additional dimensions without making any changes, since it uses a sparse vector representation. Fig. 2 represents how to transform the contextual dimensions into a prediction problem from real-valued features using Sparse Feature Vector Representation.

We used FM for two main reasons. The First reason is that the algorithm is designed to support sparse data, and the extracted contextual information will make the matrix more sparse. The second reason is that the computation cost of FM is a linear time complexity ($O(kd)$), even for additional contextual dimensions.

V. METHODOLOGY

The implementation of our model is a two-steps process. The first step is for context extraction from reviews using a custom NER and BERT model. As shown in Fig. 3, in this step we aim to switch from the two-dimensional mode used by classic Recommender Systems to multi-dimensions mode used by Context-aware Recommenders. In the second step a contextual Factorization Machine is applied to predict ratings and generate recommendations based on outputs from the previous step.

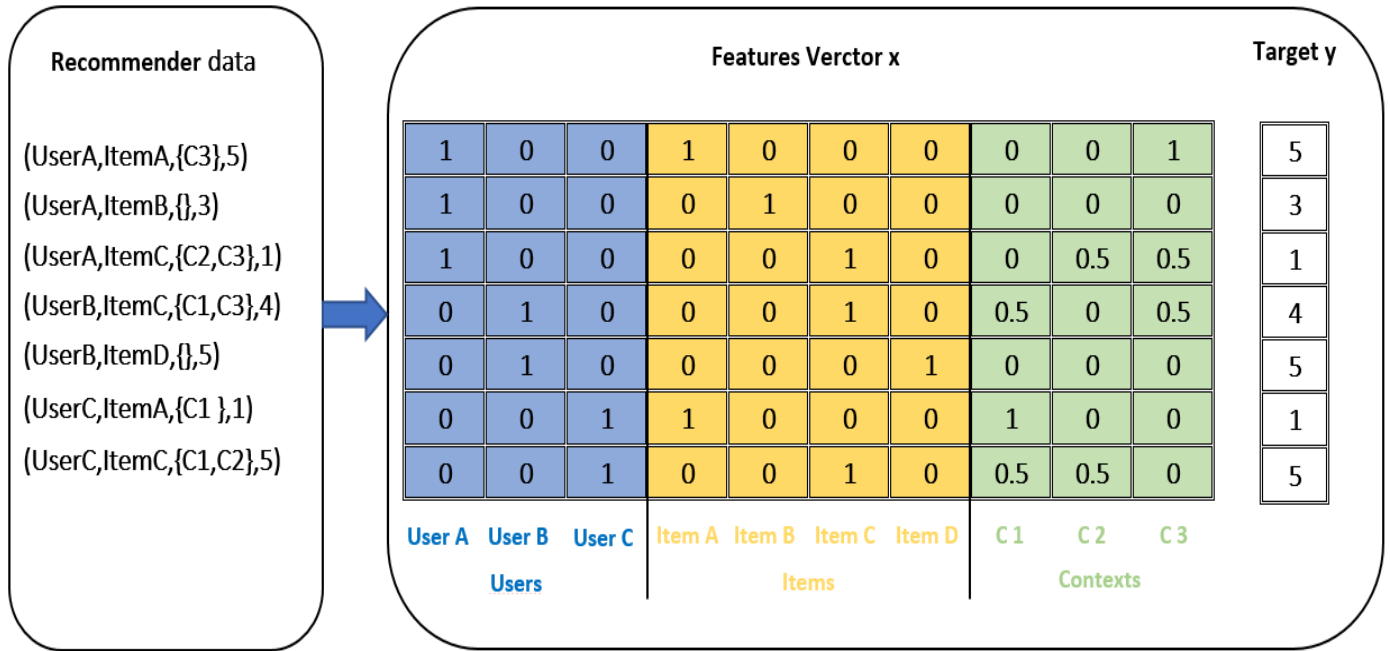


Fig. 2. Recommendation Data is Transformed to a Prediction Problem, where the First Three Columns Refer to users, the Next Four Columns Refer to Items and the Last Three Column Represent the Context.

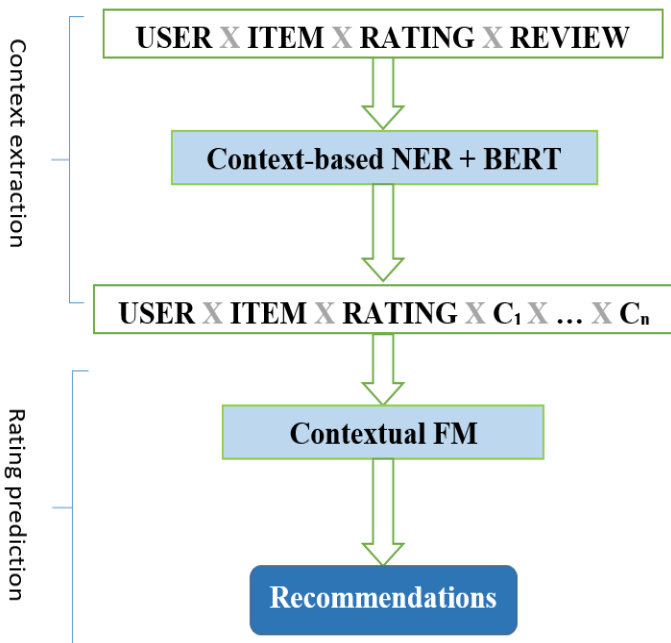


Fig. 3. The Global Architecture of our Model.

A. Context Extraction Step

In this step, The Named Entity Recognition is treated as a sequence labeling problem. Our model consists of three layers as shown in Fig. 4 namely, word embedding layer, BiLSTM layer and CRF layer. In the first layer, the BERT pre-trained model takes a sequence of n words (w_1, w_2, \dots, w_n),

then outputs a contextual embedding vector representation of each word. In contrast to context independent word embedding techniques such as Word2Vec [26], BERT is a powerful model, highly bidirectional and utilizes contextual information to learn word's context. BERT has two variants:

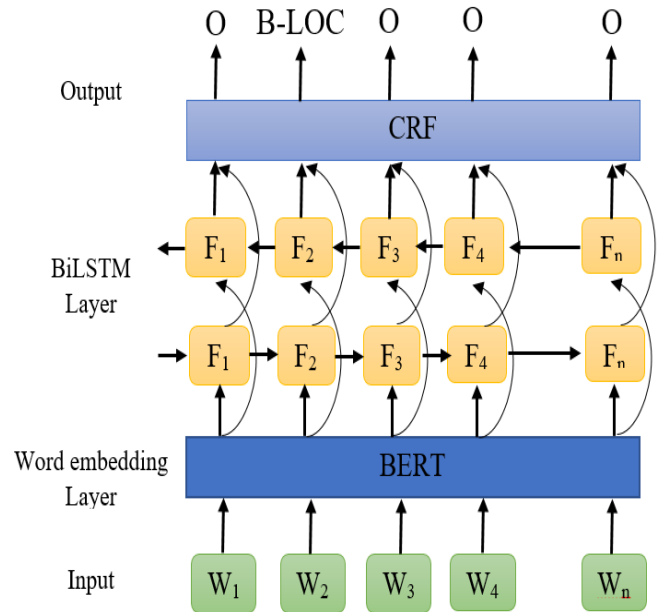


Fig. 4. The Architecture of the Custom NER.

- **BERT Base:** 12 layers (transformer encoders), 768 hidden layers, 12 attention heads and 110M parameters.

- **BERT Large**: 24 layers (transformer encoders), 1024 hidden layers, 16 attention heads and 340M parameters.

In this work we use the BERT Base model [27].

In the second layer, the Bidirectional Long Short-Term Memory (Bi-LSTM) took part. Bi-LSTM is an extension of LSTM proposed by [28] that uses forward and backward networks to process sequences. It is designed to avoid gradient vanishing and exploding and also escape the problem of long term dependency.

The output from the embedding layer is sent to the Bidirectional Long Short-Term Memory (Bi-LSTM) to extract vector features from words. Bi-LSTM concatenates the forward and the backward networks as a final result $[H_l, H_r]$. In the last layer, Conditional Random Fields (CRF) [28] outputs the most probable tag sequences. CRF is a probabilistic discriminative model that is used to label sequences. The use of CRF helps the model to learn labels and constraints that ensure the validity of the sequence. For example, the BIO format (Beginning, Inside, Outside) is a common tagging format for tagging tokens, the first word label must begin by "B" or "O" not by "I", this constraint is learned automatically by CRF.

Let X be the input sequence and Y the corresponding tag sequences, P the matrix obtained from the previous layer and T the transition matrix which represents the probability from label y_i to label y_{i+1} . The score of the tags sequence is computed as follow :

$$Score(X, Y) = \sum_{i=1}^n P_{i, y_i} + \sum_{i=1}^n T_{y_i, y_{i+1}} \quad (2)$$

B. Rating Prediction Step

In this step, a Contextual Factorization Machine (CFM) takes the output from the previous step and predicts the rating. CFM is similar to MF except that a matrix of weights is added to capture the importance of contextual dimensions. the CFM equation is given as follow:

$$y(x) = w_0 + \sum_{i=1}^d w_i b_i x_i + \sum_{i=1}^d \sum_{j=i+1}^d \langle v_i, v_j \rangle b_i b_j x_i x_j \quad (3)$$

where $w_0 \in \mathbf{R}$ is the global bias, $w \in \mathbf{R}^d$ are weights corresponding to each feature vector, $V \in \mathbf{R}^{d \times k}$ the interaction matrix, v_i i th row of the V matrix, $\langle v_i, v_j \rangle$ the interaction between the i -th and j -th variable and $B \in \mathbf{R}^p$ the matrix of weights of the importance. The parameter b_i equal to 1 for item and user and $b_i x_i$ for other dimensions.

VI. EXPERIMENT AND RESULTS

A. Corpora and Dataset

We use three corpora to pre-train our custom NER :

- **Corpus 1** is the CoNLL-2003 [29] NER dataset which consists of 18,453 sentences, 254,983 tokens and four entities namely persons, locations, organizations and miscellaneous.

- **Corpus 2** is the Groningen Meaning Bank (GMB) [30] for name entity classification, developed at the University of Groningen. It comprises 63,256 sentences, 1,388,847 tokens and eight entities (Geographical Entity, Organization, Person, Geopolitical Entity, Time, Artifact, Event, Natural Phenomenon).

- **Corpus 3** is a custom corpus that we build to face some lake in the two aforementioned corpus. In fact, after training our custom NER, it is still not able to extract some categories such as Companion context and Environmental context, e.g., "I watched the movies with my friend at the cinema" friend and cinema should be annotated as companion and location, but it's not the case. The new corpus allows us to fine-tune our custom NER, and tackle this problem. It is created in a BIO(Beginning, Inside, Outside) format and consists of 3500 sentences, 43,565 tokens and three entities.

To evaluate our model, we have selected Amazon Customer Reviews Dataset [31] and Yelp dataset [32]. Amazon dataset consists of customer reviews, ratings and product metadata(price, brand, descriptions, ...). It includes more than 233.1M reviews collected between 1996 and 2018 and 21 categories of products. This dataset is considered as the largest public dataset for rating. The Yelp dataset is a free to use dataset for academic and personal purposes, it contains more than 8.5M reviews and more than 160K businesses.

We adopted three metrics to evaluate our custom NER namely Precision, F1 and Recall:

$$Precision(P) = \frac{TP}{TP + FP} \quad (4)$$

$$Recall(R) = \frac{TP}{TP + FN} \quad (5)$$

$$Precision(P) = \frac{Precision \times Recall}{Precision + Recall} \quad (6)$$

And we use the Mean Square Error (MSE) to evaluate the performance of the CFM algorithm. The corresponding equation of the MSE is introduced as follow:

$$MSE = \frac{1}{N} \sum_{i=1}^n (r_n - \hat{r})^2 \quad (7)$$

B. Experimental Setting

As previously mentioned the proposed approach consists of two steps: the context extraction step and the rating prediction step. In the first step, the custom NER is trained using the FLAIR library, an open source NLP framework for state-of-the-art text classification and sequence labeling. We use Google Colab to train the model (GPU 12GB). In order to not exceed the available GPU memory, we fix the mini-batch to 32 and the maximum sequence length 512. We use a Bi-LSTM with a single layer with a hidden size of 256 to process input sequences and a learning rate to 0.1.

In the second step, We build our CFM using Tensorflow framework [33], our implementation is inspired from [34][35]. We split the data into 80% for training and 20% for test. As we are dealing with a regression problem, the model

parameters are learned by minimizing the loss function, we also prevent overfitting by adding a L2 regularization term. Since it is efficient with sparse data, we use a gradient-based optimizer. We fix the number of iteration to 1000 since the CFM algorithm needs more time to converge.

C. Results and Discussion

1) Custom NER: Results of the custom NER for different corpus are represented in Fig. 5. As we can see, the custom NER achieves the best F1 score of 91.59% for Corpus1, 89.92% for Corpus3 and the worst F1 score of 80.17% for Corpus2. The disparity in results can be explained by the fact that the quality of data from Corpus to another. The GMB Corpus gets the worst performance because it is not perfect. It is not completely human, the corpus is built using existing annotation and must be corrected manually by humans.

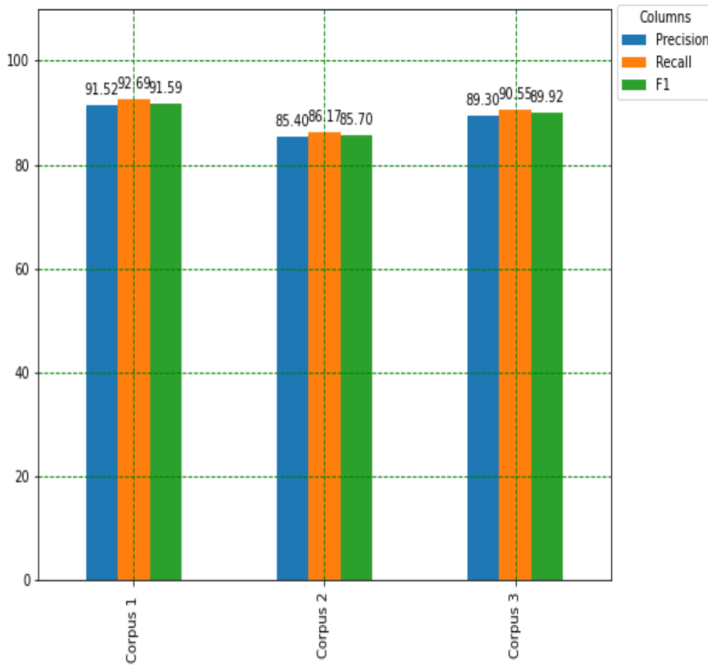


Fig. 5. Results of Three Corpus using F1 Score, Precision and Recall.

Table II shows the obtained results for each entity. The model gets a F1 score of 94.34% for Companion, 90.70% for location, 85.90% for environmental entity and 77.04% for Time. We can explain the difference in obtained results by the difference in the degree of difficulty encountered during the extraction of each entity. For instance it is less difficult to extract location (i.e., name of city) from unstructured text than extract time entity which can be written in different ways (i.e, 5pm, 5p.m, 5 p.m, Five pm ,). Even ContextNER does get good performance on Corpus 2 it still competitive compared with other recent models.

2) Contextual Factorization Machine: To validate the effectiveness of CFM we compare it with five competitive baselines:

- **FM [25]:** Factorization Machine is the famous Collaborative filtering RS method. It takes the matrix of rating and transforms it to two low-rank matrices.

TABLE II. OBTAINED RESULTS FOR EACH ENTITY

Entity	Precision	Recall	F1 score
Location	91.47%	87.23%	90.70%
Time	77.74%	76.35%	77.04%
Companion	92.59%	96.15%	94.34%
Environmental	86.37%	85.43%	85.90%

- **PMF [36]:** Probabilistic Matrix Factorization uses Gaussian distributions to estimate users and items latent factors.

- **HFT [37]:** Hidden Factor as Topic extracts latent factors from reviews by employing topic distributions.

- **CDL [38]:** Collaborative Deep Learning is a hierarchical Bayesian model which performs deep representation learning for Collaborative filtering RS.

- **DeepCoNN [3]:** Deep Cooperative Neural Networks model extracts user behaviours and item properties from reviews.

All works already mentioned have incorporated reviews except FM and PMF.

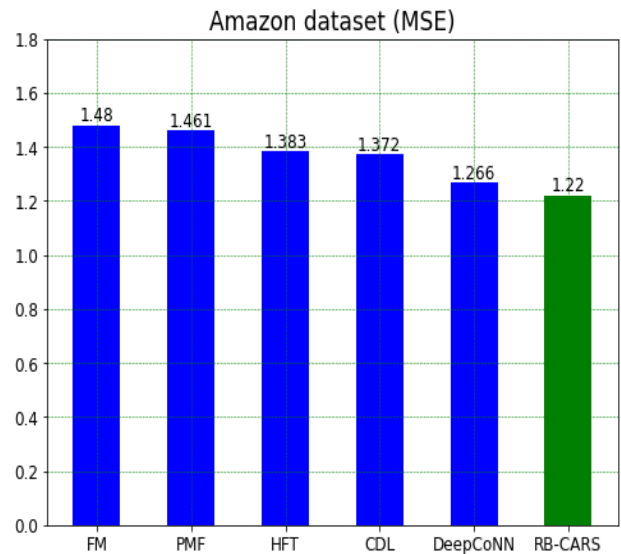


Fig. 6. Obtained Results for Amazon Dataset.

In Fig. 6, results for the Amazon dataset are reported in terms of MSE. We can definitely notice that our model significantly improves the quality of recommendation passing from 1.480 for FM to 1.22 which means an improvement of 12.9% over FM model and 5.3% over DeepCoNN.

By the same way, our model shows good performance over the other models for the Yelp dataset. As we can see in Fig. 7, the model gets a MSE 1.382 over 1.792 for FM model

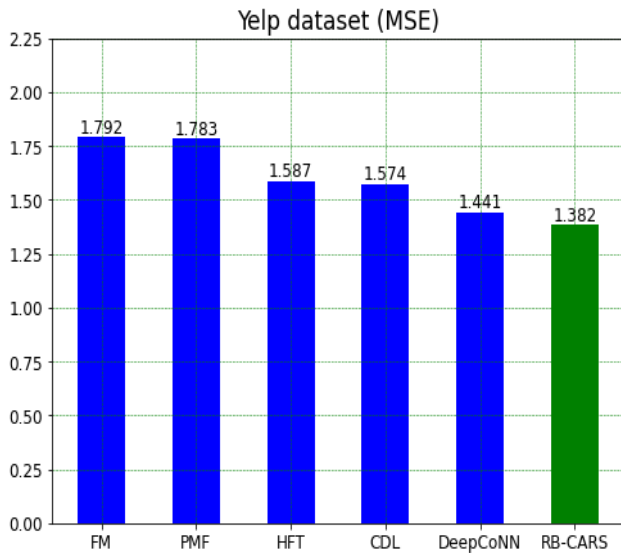


Fig. 7. Obtained Results for Yelp Dataset.

and 1.441 for DeepCoNN, equivalent to 13.4% and 4.9% of improvement respectively for both models.

TABLE III. OBTAINED RESULTS FOR EACH ENTITY

Datasets	Amazon	Yelp	All datasets average
FM	1.480	1.792	1.636
PMF	1.461	1.783	1.622
HFT	1.383	1.587	1.485
CDL	1.372	1.574	1.473
DeepCoNN	1.266	1.441	1.353
DeepCoNN	1.220	1.382	1.301
Improvement (%)	12.9	13.4	13.15

The difference in results is mainly related to the sparsity of the two datasets and it is worth to note that with other datasets less sparse than the two datasets used in these experiments our model can perform better. Table III represents all obtained results.

VII. CONCLUSION

This article presented an automatic method to extract contextual information from users' reviews, then use it to improve recommendation quality with less time spent in feature engineering. This work is divided into two main steps:

The first for context extraction using a custom NER, where the model used in this step consists of three layers, namely

word embedding layer which takes a sequence of words and outputs a contextual embedding vector representation of each word using BERT model, the Bi-LSTM layer which extracts vector features from words generated by the previous layer and the last layer called the CRF layer which helps to automatically learn labels and constraints and guarantee the sequence validity. The second step is for ratings prediction. The CFM algorithm takes the output from the first step and computes the ratings. In contrast to the generic FM, the CFM is able to capture the importance of the contextual dimensions and incorporate them into the process of recommendation.

To evaluate the performance, the proposed model was compared with five models and obtained results show that the model achieves good results. For future work, the proposed model will be improved in both steps, namely the data extraction step and techniques utilized for this end and also for ratings prediction.

REFERENCES

- [1] "GroupLens: applying collaborative filtering to Usenet news: Communications of the ACM: Vol 40, No 3." <https://dl.acm.org/doi/10.1145/245108.245126>.
- [2] U. Panniello, A. Tuzhilin, and M. Gorgoglione, "Comparing context-aware recommender systems in terms of accuracy and diversity," *User Model User-Adap Inter*, vol. 24, no. 1, pp. 35–65, Feb. 2014, doi: 10.1007/s11257-012-9135-y.
- [3] L. Zheng, V. Noroozi, and P. S. Yu, "Joint Deep Modeling of Users and Items Using Reviews for Recommendation," arXiv:1701.04783 [cs], Jan. 2017, Accessed: Nov. 11, 2021. [Online]. Available: <http://arxiv.org/abs/1701.04783>.
- [4] R. Catherine and W. Cohen, "TransNets: Learning to Transform for Recommendation," in *Proceedings of the Eleventh ACM Conference on Recommender Systems*, New York, NY, USA, Aug. 2017, pp. 288–296. doi: 10.1145/3109859.3109878.
- [5] J. McAuley and J. Leskovec, "Hidden factors and hidden topics: understanding rating dimensions with review text," in *Proceedings of the 7th ACM conference on Recommender systems*, New York, NY, USA, Oct. 2013, pp. 165–172. doi: 10.1145/2507157.2507163.
- [6] Y. Tan, M. Zhang, Y. Liu, and S. Ma, "Rating-Boosted Latent Topics: Understanding Users and Items with Ratings and Reviews," 2016.
- [7] Y. Zhang, G. Lai, M. Zhang, Y. Zhang, Y. Liu, and S. Ma, "Explicit factor models for explainable recommendation based on phrase-level sentiment analysis," in *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, New York, NY, USA, Jul. 2014, pp. 83–92. doi: 10.1145/2600428.2609579.
- [8] A. Levi, O. Mokryn, C. Diot, and N. Taft, "Finding a needle in a haystack of reviews: cold start context-based hotel recommender system," in *Proceedings of the sixth ACM conference on Recommender systems*, New York, NY, USA, Sep. 2012, pp. 115–122. doi: 10.1145/2365952.2365977.
- [9] S. Aciar, "Mining Context Information from Consumer's Reviews." <https://www.semanticscholar.org/paper/Mining-Context-Information-from-Consumer-%E2%80%99-s-Aciar/3c0977e396849c345ff5283772ecd5c1b7097c08>.
- [10] N. Hariri, B. Mobasher, R. Burke, and Y. Zheng, "Context-Aware Recommendation Based On Review Mining," 2011.
- [11] P. Campos, N. Rodríguez-Artigot, and I. Cantador, "Extracting Context Data from User Reviews for Recommendation: A Linked Data Approach," 2017.
- [12] F. Z. Lahlou, H. Benbrahim, and I. Kassou, "Review Aware Recommender System: Using Reviews for Context Aware Recommendation," *IJDAI*, vol. 10, no. 2, pp. 28–50, Jul. 2018, doi: 10.4018/IJDAI.2018070102.
- [13] G. Adomavicius, B. Mobasher, F. Ricci, and A. Tuzhilin, "Context-Aware Recommender Systems," *AI Magazine*, vol. 32, no. 3, Art. no. 3, 2011, doi: 10.1609/aimag.v32i3.2364.

- [14] G. Castelli, A. Rosi, M. Mamei, and F. Zambonelli, "A Simple Model and Infrastructure for Context-Aware Browsing of the World," in Fifth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom'07), Mar. 2007, pp. 229–238. doi: 10.1109/PERCOM.2007.4.
- [15] J. D. Kim, J. Son, and D. K. Baik, "CA 5W1H Onto: Ontological context-aware model based on 5W1H," International Journal of Distributed Sensor Networks, vol. 2012, no. 247346, 2012, doi: 10.1155/2012/247346.
- [16] T. Chaari, E. Dejene, F. Laforest, and V.-M. Scuturici, "A Comprehensive Approach to Model and Use Context for Adapting Applications in Pervasive Environments," Int. Journal of Systems and software, vol. 80, pp. 1973–1992, Dec. 2007, doi: 10.1016/j.jss.2007.03.010.
- [17] H. Chen, T. Finin, and A. Joshi, "The SOUPA Ontology for Pervasive Computing," in Ontologies for Agents: Theory and Experiences, Basel, 2005, pp. 233–258. doi: 10.1007/3-7643-7361-X10.
- [18] B. Y. Lin, F. Xu, Z. Luo, and K. Zhu, "Multi-channel BiLSTM-CRF Model for Emerging Named Entity Recognition in Social Media," in Proceedings of the 3rd Workshop on Noisy User-generated Text, Copenhagen, Denmark, Sep. 2017, pp. 160–165. doi: 10.18653/v1/W17-4421.
- [19] H. Yan, B. Deng, X. Li, and X. Qiu, "TENER: Adapting Transformer Encoder for Named Entity Recognition," arXiv:1911.04474 [cs], Dec. 2019, Accessed: Nov. 14, 2021. [Online]. Available: <http://arxiv.org/abs/1911.04474>
- [20] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.
- [21] Z. Huang, W. Xu, and K. Yu, "Bidirectional LSTM-CRF Models for Sequence Tagging," arXiv:1508.01991 [cs], Aug. 2015, Accessed: Nov. 14, 2021. [Online]. Available: <http://arxiv.org/abs/1508.01991>
- [22] A. Radford and K. Narasimhan, "Improving Language Understanding by Generative Pre-Training," undefined, 2018, Accessed: Nov. 14, 2021. [Online]. Available: <https://www.semanticscholar.org/paper/Improving-Language-Understanding-by-Generative-Radford-Narasimhan/cd18800a0fe0b668a1cc19f2ec95b5003d0a5035>
- [23] M. E. Peters et al., "Deep contextualized word representations," arXiv:1802.05365 [cs], Mar. 2018, Accessed: Nov. 14, 2021. [Online]. Available: <http://arxiv.org/abs/1802.05365>
- [24] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," arXiv:1810.04805 [cs], May 2019, Accessed: Nov. 14, 2021. [Online]. Available: <http://arxiv.org/abs/1810.04805>
- [25] S. Rendle, "Factorization Machines," in 2010 IEEE International Conference on Data Mining, Dec. 2010, pp. 995–1000. doi: 10.1109/ICDM.2010.127.
- [26] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," arXiv:1310.4546 [cs, stat], Oct. 2013, Accessed: Nov. 16, 2021. [Online]. Available: <http://arxiv.org/abs/1310.4546>
- [27] Wolf et al., "HuggingFace's Transformers: State-of-the-art Natural Language Processing," arXiv:1910.03771 [cs], Jul. 2020, Accessed: Nov. 16, 2021. [Online]. Available: <http://arxiv.org/abs/1910.03771>
- [28] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data," in Proceedings of the Eighteenth International Conference on Machine Learning, Jun. 2001, pp. 282–289.
- [29] E. F. T. K. Sang and F. De Meulder, "Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition," arXiv:cs/0306050, Jun. 2003, Accessed: Nov. 20, 2021. [Online]. Available: <http://arxiv.org/abs/cs/0306050>
- [30] "Home - Groningen Meaning Bank." <https://gmb.let.rug.nl/> (accessed Nov. 20, 2021).
- [31] "Amazon review data." <https://jmcauley.ucsd.edu/data/amazon/> (accessed Nov. 20, 2021).
- [32] "Yelp Dataset." <https://www.yelp.com/dataset> (accessed Nov. 20, 2021).
- [33] M. Abadi et al., "TensorFlow: A system for large-scale machine learning," arXiv:1605.08695 [cs], May 2016, Accessed: Nov. 20, 2021. [Online]. Available: <http://arxiv.org/abs/1605.08695>
- [34] S. Rendle, "Factorization Machines with libFM," ACM Trans. Intell. Syst. Technol., vol. 3, no. 3, p. 57:1–57:22, May 2012, doi: 10.1145/2168752.2168771.
- [35] F. Z. Lahlou, Textual context Aware Factorization Machines. 2019. Accessed: Nov. 20, 2021. [Online]. Available: <https://github.com/fzlahlou/TCAFM/blob/2beb2e73a28f2f577c05d4d77122bf26225e0dd1/TCAFM%20and%20FM%20Code.py>
- [36] R. Salakhutdinov and A. Mnih, "Probabilistic Matrix Factorization," in Proceedings of the 20th International Conference on Neural Information Processing Systems, Red Hook, NY, USA, Dec. 2007, pp. 1257–1264.
- [37] J. McAuley and J. Leskovec, "Hidden factors and hidden topics: understanding rating dimensions with review text," in Proceedings of the 7th ACM conference on Recommender systems, New York, NY, USA, Oct. 2013, pp. 165–172. doi: 10.1145/2507157.2507163.
- [38] H. Wang, N. Wang, and D.-Y. Yeung, "Collaborative Deep Learning for Recommender Systems," arXiv:1409.2944 [cs, stat], Jun. 2015, Accessed: Nov. 22, 2021. [Online]. Available: <http://arxiv.org/abs/1409.2944>