COUPLENET: Paying Attention to Couples with Coupled Attention for Relationship Recommendation

Yi Tay,[1] **Luu Anh Tuan,**[2] **Siu Cheung Hui**[3]
[1, 3] Nanyang Technological University
School of Computer Science and Engineering, Singapore
[2] Institute for Infocomm Research, Singapore

Abstract

Dating and romantic relationships not only play a huge role in our personal lives but also collectively influence and shape society. Today, many romantic partnerships originate from the Internet, signifying the importance of technology and the web in modern dating. In this paper, we present a text-based computational approach for estimating the relationship compatibility of two users on social media. Unlike many previous works that propose reciprocal recommender systems for online dating websites, we devise a distant supervision heuristic to obtain real world couples from social platforms such as Twitter. Our approach, the COUPLENET is an end-to-end deep learning based estimator that analyzes the social profiles of two users and subsequently performs a similarity match between the users. Intuitively, our approach performs both user profiling and match-making within a unified end-to-end framework. COUPLENET utilizes hierarchical recurrent neural models for learning representations of user profiles and subsequently coupled attention mechanisms to fuse information aggregated from two users. To the best of our knowledge, our approach is the first data-driven deep learning approach for our novel relationship recommendation problem. We benchmark our COUPLENET against several machine learning and deep learning baselines. Experimental results show that our approach outperforms all approaches significantly in terms of precision. Qualitative analysis shows that our model is capable of also producing explainable results to users.

Introduction

The social web has become a common means for seeking romantic companionship, made evident by the wide assortment of online dating sites that are available on the Internet. As such, the notion of relationship recommendation systems is not only interesting but also highly applicable. This paper investigates the possibility and effectiveness of a deep learning based relationship recommendation system. An overarching research question is whether modern artificial intelligence (AI) techniques, given social profiles, can successfully approximate successful relationships and measure the relationship compatibility of two users.

Prior works in this area (Xia et al. 2015; 2014; Krzywicki et al. 2014; Xia et al. 2015) have been mainly considered the 'online dating recommendation' problem, i.e., focusing on the reciprocal domain of dating social networks (DSN) such as Tinder and OKCupid. While the functionality and mechanics of dating sites differ across the spectrum, the main objective is usually to facilitate communication between users, who are explicitly seeking relationships. Another key characteristic of many DSNs is the functionality that enables a user to express interest to another user, e.g., swiping right on Tinder. Therefore, many of prior work in this area focus on reciprocal recommendation, i.e., predicting if two users will *like* or *text* each other. Intuitively, we note that likes and replies on DSNs are not any concrete statements of compatibility nor evidence of any long-term relationship. For instance, a user may have many reciprocal matches on Tinder but eventually form meaningful friendships or relationships with only a small fraction.

Our work, however, focuses on a seemingly similar but vastly different problem. Instead of relying on reciprocal signals from DSNs, our work proposes a novel distant supervision scheme, constructing a dataset of real world couples from regular[1] social networks (RSN). Our distant supervision scheme is based on Twitter, searching for tweets such as *'good night baby love you '* and *'darling i love you so much '* to indicate that two users are in a stable and loving relationship (at least at that time). Using this labeled dataset, we train a distant supervision based learning to rank model to predict relationship compatibility between two users using their social profiles. The key idea is that social profiles contain cues pertaining to personality and interests that may be a predictor if whether two people are romantically compatible. Moreover, unlike many prior works that operate on propriety datasets (Xia et al. 2014; Krzywicki et al. 2014; Xia et al. 2015), our dataset is publicly and legally obtainable via the official Twitter API. In this work, we construct the first public dataset of approximately 2 million tweets for the task of relationship recommendation.

Another key advantage is that our method trains on regular social networks, which spares itself from the inherent problems faced by DSNs, e.g., deceptive self-presentation, harassment, bots, etc. (Masden and Edwards 2015). More specifically, self-presented information on DSNs might be

[1] We define regular social networks (RSN) as any social network that is not primarily a DSN, e.g., Facebook, Twitter.

inaccurate with the sole motivation of appearing more attractive (Toma and Hancock 2010; Hancock, Toma, and Ellison 2007). In our work, we argue that measuring the compatibility of two users on RSN might be more suitable, eliminating any potential explicit self-presentation bias. Intuitively, social posts such as tweets can reveal information regarding personality, interests and attributes (Arnoux et al. 2017; Wei et al. 2017).

Finally, we propose COUPLENET, an end-to-end deep learning based architecture for estimating the compatibility of two users on RSNs. COUPLENET takes the social profiles of two users as an input and computes a compatibility score. This score can then be used to serve a ranked list to users and subsequently embedded in some kind of 'who to follow' service. COUPLENET is characterized by its Coupled Attention, which learns to pay attention to parts of a user's profile dynamically based on the current candidate user. COUPLENET also does not require any feature engineering and is a proof-of-concept of a completely text-based relationship recommender system. Additionally, COUPLENET is also capable of providing explainable recommendations which we further elaborate in our qualitative experiments.

Our Contributions

This section provides an overview of the main contributions of this work.

- We propose a novel problem of *relationship recommendation* (RSR). Different from the reciprocal recommendation problem on DSNs, our RSR task operates on regular social networks (RSN), estimating long-term and serious relationship compatibility based on social posts such as tweets.
- We propose a novel distant supervision scheme to construct the first publicly available (distributable in the form of tweet ids) dataset for the RSR task. Our dataset, which we call the LOVEBIRDS2M dataset consists of approximately 2 million tweets.
- We propose a novel deep learning model for the task of RSR. Our model, the COUPLENET uses hierarchical Gated Recurrent Units (GRUs) and coupled attention layers to model the interactions between two users. To the best of our knowledge, this is the first deep learning model for both RSR and reciprocal recommendation problems.
- We evaluate several strong machine learning and neural baselines on the RSR task. This includes the recently proposed DeepCoNN (*Deep Co-operative Neural Networks*) (Zheng, Noroozi, and Yu 2017) for item recommendation. COUPLENET significantly outperforms DeepCoNN with a 200% relative improvement in precision metrics such as Hit Ratio (HR@N). Overall findings show that a text-only deep learning system for RSR task is plausible and reasonably effective.
- We show that COUPLENET produces explainable recommendation by analyzing the attention maps of the coupled attention layers.

Related Work

In this section, we review existing literature that is related to our work.

Reciprocal and Dating Recommendation

Prior works on online dating recommendation (Xia et al. 2015; Tu et al. 2014; Krzywicki et al. 2014; Akehurst et al. 2011) mainly focus on designing systems for dating social networks (DSN), i.e., websites whereby users are on for the specific purpose of finding a potential partner. Moreover, all existing works have primarily focused on the notion of reciprocal relationships, e.g., a successful signal implied a two way signal (likes or replies) between two users.

Tu et al. (Tu et al. 2014) proposed a recommendation system based on Latent Dirichlet Allocation (LDA) to match users based on messaging and conversational history between users. Xia et al. (Xia et al. 2015; 2014) cast the dating recommendation problem into a link prediction task, proposing a graph-based approach based on user interactions. The CCR (Content-Collaborative Reciprocal Recommender System) (Akehurst et al. 2011) was proposed by Akehurtst et al. for the task of reciprocal recommendation, utilizing content-based features (user profile similarity) and collaborative filtering features (user-user interactions). However, all of their approaches operate on a propriety dataset obtained via collaboration with online dating sites. This hinders research efforts in this domain.

Our work proposes a different direction from the standard reciprocal recommendation (RR) models. The objective of our work is fundamentally different, i.e., instead of finding users that might reciprocate to each other, we learn to functionally approximate the essence of a good (possibly stable and serious) relationship, learning a compatibility score for two users given their regular social profiles (e.g., Twitter). To the best of our knowledge, our work is the first to build a relationship recommendation model based on a distant supervision signal on real world relationships. Hence, we distinguish our work from all existing works on online dating recommendation.

Moreover, our dataset is obtained legally via the official twitter API and can be distributed for future research. Unlike prior work (Xia et al. 2015) which might invoke privacy concerns especially with the usage of conversation history, the users employed in our study have public twitter feeds. We note that publicly available twitter datasets have been the cornerstone of many scientific studies especially in the fields of social science and natural language processing (NLP).

Across scientific literature, several other aspects of online dating have been extensively studied. Nagarajan and Hearst (Nagarajan and Hearst 2009) studied self-presentation on online dating sites by specifically examining language on dating profiles. Hancock et al. presented an analysis on deception and lying on online dating profiles (Hancock, Toma, and Ellison 2007), reporting that at least 50% of participants provide deceptive information pertaining to physical attributes such as height, weight or age. Toma et al. (Toma and Hancock 2010) investigated the correlation between linguistic cues and deception on online dating profiles. Maldeniya et al. (Maldeniya et al. 2017) studied how textual similarity between user profiles impacts the likelihood of reciprocal behavior. A recent work by Cobb and Kohno (Cobb and Kohno) provided an extensive study which tries to understand users privacy preferences and practices in online dating.

User Profiling and Friend Recommendation

Our work is a cross between user profiling and user matchmaking systems. An earlier work, (Diaz, Metzler, and Amer-Yahia) proposed a gradient-boosted learning-to-rank model for match-making users on a dating forum. While the authors ran experiments on a dating service website, the authors drew parallels with other match-making services such as job-seeking forums. The user profiling aspect in our work comes from the fact that we use social networks to learn user representations. As such, our approach performs both user profiling and then match-making within an end-to-end framework. (Wei et al. 2017) proposed a deep learning personality detection system which is trained on social posts on Weibo and Twitter. (Arnoux et al. 2017) proposed a Twitter personality detection system based on machine learning models. (Benton, Arora, and Dredze 2016) learned multiview embeddings of Twitter users using canonical correlation analysis for friend recommendation. From an application perspective, our work is also highly related to 'People you might know' or 'who to follow' (WTF) services on RSNs (Gupta et al. 2013) albeit taking a romantic twist. In practical applications, our RSN based relationship recommender can either be deployed as part of a WTF service, or to increase the visibility of the content of users with high compatibility score.

Deep Learning and Collaborative Ranking

One-class collaborative filtering (also known as collaborative ranking) (Hu, Koren, and Volinsky 2008) is a central research problem in IR. In general, deep learning (He et al. 2017; Tay, Luu, and Hui 2017) has also been recently very popular for collaborative ranking problems today. However, to the best of our knowledge, our work is the first deep learning based approach for the online dating domain. Our approach also follows the neural IR approach which is mainly concerned with modeling document-query pairs (Severyn and Moschitti 2015; Tay et al. 2017) or user-item pairs (Zheng, Noroozi, and Yu 2017; Tay, Tuan, and Hui 2018) since we deal with the textual domain. Finally, our work leverages recent advances in deep learning, namely Gated Recurrent Units (Cho et al. 2014) and Neural Attention (Yang et al. 2016; Luong, Pham, and Manning 2015; Bahdanau, Cho, and Bengio 2014). The key idea of neural attention is to learn to attend to various segments of a document, eliminating noise and emphasizing the important segments for prediction.

Problem Definition and Notation

In this section, we introduce the formal problem definition of this work.

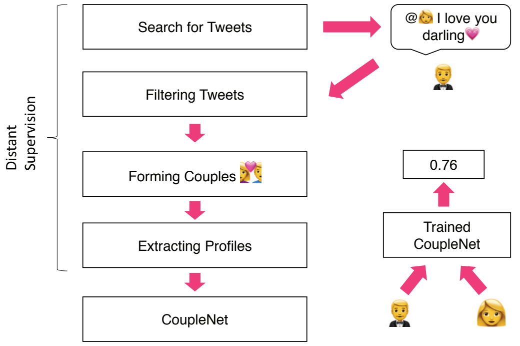


Figure 1: Overview of our distant supervision and deep learning approach for relationship recommendation.

Definition 0.1. *Let U be the set of Users. Let s_i be the social profile of user i which is denoted by $u_i \in U$. Each social profile $s_i \in S$ contains η documents. Each document $d_i \in s_i$ contains a maximum of L words. Given a user u_i and his or her social profile s_i, the task of the Relationship Recommendation problem is to produce a ranked list of candidates based on a computed relevance score $F(s_i, s_j)$ where s_j is the social profile of the candidate user u_j. $F(.)$ is a parameterized function.*

There are mainly three types of learning to rank methods, namely pointwise, pairwise and list-wise. Pointwise considers each user pair individually, computing a relevance score solely based on the current sample, i.e., binary classification. Pairwise trains via noise constrastive estimation, which often minimizes a loss function like the margin based hinge loss. List-wise considers an entire list of candidates and is seldom employed due to the cumbersome constraints that stem from implementation efforts. Our proposed COUPLENET employs a pairwise paradigm. The intuition for this is that, relationship recommendation is considered very sparse and has very imbalanced classes (for each user, only one ground truth exists). Hence, training binary classification models suffers from class imbalance. Moreover, the good performance of pairwise learning to rank is also motivated by our early experiments.

The Love Birds Dataset

Since there are no publicly available datasets for training relationship recommendation models, we construct our own. The goal is to construct a list of user pairs in which both users are in relationship. Our dataset is constructed via distant supervision from Twitter. We call this dataset the *Love Birds* dataset. This not only references the metaphorical meaning of the phrase 'love birds' but also deliberately references the fact that the Twitter icon is a bird. This section describes the construction of our dataset[2]. Figure 1 describes the overall process of our distant supervision framework.

[2] To facilitate further research, our dataset will be released at https://github.com/vanzytay/ICWSM18_LB2M. Distribution will come in the form of tweet IDs and labels, to adhere to the regulations of the Twitter public API.

Distant Supervision

Using the Twitter public API, we collected tweets with emojis contains the keyword *'heart'* in its description. The key is to find tweets where a user expresses love to another user. We observed that there are countless tweets such as *'good night baby love you '* and *'darling i love you so much '* on Twitter. As such, the initial list of tweets is crawled by watching heart and love-related emojis, e.g., , , etc. By collecting tweets containing these emojis, we form our initial candidate list of couple tweets (tweets in which two people in a relationship send to each other). Through this process, we collected 10 million tweets over a span of a couple of days. Each tweet will contain a sender and a target (the user mentioned and also the target of affection).

Keyword Filtering We also noticed that the love related emojis do not necessarily imply a romantic relationship between two users. For instance, we noticed that a large percentage of such tweets are affection towards family members. Given the large corpus of candidates, we can apply a stricter filtering rule to obtain true couples. To this end, we use a ban list of words such as 'bro', 'sis', 'dad', 'mum' and apply regular expression based filtering on the candidates. We also observed a huge amount of music related tweets, e.g., 'I love this song so much !'. Hence, we also included music-related keywords such as 'perform', 'music', 'official' and 'song'. Finally, we also noticed that people use the heart emoji frequently when asking for someone to follow them back. As such, we also ban the word 'follow'.

User-based Filtering We further restricted tweets to contain only a single mention. Intuitively, mentioning more than one person implies a group message rather than a couple tweet. We also checked if one user has a much higher follower count over the other user. In this case, we found that this is because people send love messages to popular pop idols (we found that a huge bulk of crawled tweets came from fangirls sending love message to @harrystylesofficial). Any tweet with a user containing more than 5K followers is being removed from the candidate list.

Forming Couple Pairs

Finally, we arrive at 12K tweets after aggressive filtering. Using the 12K 'cleaned' couple tweets, we formed a list of couples. We sorted couples in alphabetical order, i.e., (clara, ben) becomes (ben, clara) and removed duplicate couples to ensure that there are no 'bidirectional' pairs in the dataset. For each user on this list, we crawled their timeline and collected 200 latest tweets from their timeline. Subsequently, we applied further preprocessing to remove explicit couple information. Notably, we do not differentiate between male and female users (since twitter API does not provide this information either). The signal for distant supervision can be thought of as an explicit signal which is commonplace in recommendation problems that are based on explicit feedback (user ratings, reviews, etc.). In this case, an act (tweet) of love / affection is the signal used. We call this explicit couple information.

Removing Additional Explicit Couple Information To ensure that there are no *additional* explicit couple information in each user's timeline, we removed all tweets with any words of affection (heart-related emojis, 'love', 'dear', etc.). We also masked all mentions with the @USER symbol. This is to ensure that there is no explicit leak of signals in the final dataset. Naturally, a more accurate method is to determine the date in which users got to know each other and then subsequently construct timelines based on tweets prior to that date. Unfortunately, there is no automatic and trivial way to easily determine this information. Consequently, a fraction of their timeline would possibly have been tweeted when the users have already been together in a relationship. As such, in order to remove as much 'couple' signals, we try our best to mask such information.

Why Twitter?

Finally, we answer the question of why Twitter was chosen as our primary data source. One key desiderata was that the data should be public, differentiating ourselves from other works that use proprietary datasets (Xia et al. 2015; Tu et al. 2014). In designing our experiments, we considered two other popular social platforms, i.e., Facebook and Instagram. Firstly, while Facebook provides explicit relationship information, we found that there is a lack of personal, personality-revealing posts on Facebook. For a large majority of users, the only signals on Facebook mainly consist of shares and likes of articles. The amount of original content created per user is extremely low compared to Twitter whereby it is trivial to obtain more than 200 tweets per user. Pertaining to Instagram, we found that posts are also generally much sparser especially in regards to frequency, making it difficult to amass large amounts of data per user. Moreover, Instagram adds a layer of difficulty as Instagram is primarily multi-modal. In our Twitter dataset, we can easily mask explicit couple information by keyword filters. However, it is non-trivial to mask a user's face on an image. Nevertheless, we would like to consider Instagram as an interesting line of future work.

Dataset Statistics

Our final dataset consists of 1.858M tweets (200 tweets per user). The total number of users is 9290 and 4645 couple pairs. The couple pairs are split into training, testing and development with a 80/10/10 split. The total vocabulary size (after lowercasing) is 2.33M. Ideally, more user pairs could be included in the dataset. However, we also note that the dataset is quite large (almost 2 million tweets) already, posing a challenge for standard hardware with mid-range graphic cards. Since this is the first dataset created for this novel problem, we leave the construction of a larger benchmark for future work.

Our Proposed Approach

In this section, we introduce our deep learning architecture - the COUPLENET. Overall, our neural architecture is a hierarchical recurrent model (Yang et al. 2016), utilizing multi-layered attentions at different hierarchical levels. An

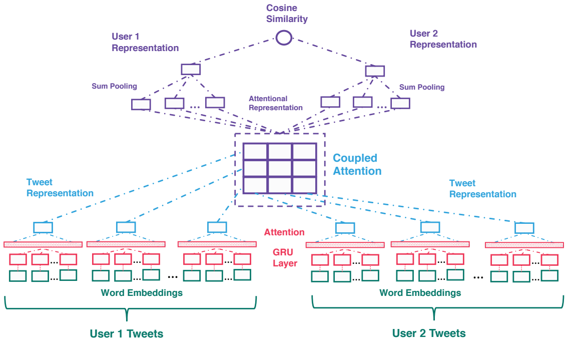


Figure 2: Overview of COUPLENET model architecture illustrating the computation of similarity score for User 1 and User 2. Negative sampling side of the network is omitted due to lack of space.

overview of the model architecture is illustrated in Figure 2. There are two sides of the network, one for each user. Our network follows a 'Siamese' architecture, with shared parameters for each side of the network. A single data input to our model comprises user pairs $(U1, U2)$ (couples) and $(U1, U3)$ (negative samples). Each user has K tweets each with a maximum length of L. The value of K and L are tunnable hyperparameters.

Embedding Layer

For each user, the inputs to our network are a matrix of indices, each corresponding to a specific word in the dictionary. The embedding matrix $\mathbf{W} \in \mathbb{R}^{d \times |V|}$ acts as a look-up whereby each index selects a d dimensional vector, i.e., the word representation. Thus, for each user, we have $K \times L$ vectors of dimension size d. The embedding layer is shared for all users and is initialized with pretrained word vectors.

Learning Tweet Representations

For each user, the output of the embedding layer is a tensor of shape $K \times L \times d$. We pass each tweet through a recurrent neural network. More specifically, we use Gated Recurrent Units (GRU) encoders with attentional pooling to learn a n dimensional vector for each tweet.

Gated Recurrent Units (GRU) The GRU accepts a sequence of vectors and recursively composes each input vector into a hidden state. The recursive operation of the GRU is defined as follows:

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z)$$
$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r)$$
$$\hat{h_t} = tanh(W_h\, x_t + U_h(r_t h_{t-1}) + b_h)$$
$$h_t = z_t\, h_{t-1} + (1 - z_t)\, \hat{h_t}$$

where h_t is the hidden state at time step t, z_t and r_t are the update gate and reset gate at time step t respectively. σ is the sigmoid function. x_t is the input to the GRU unit at time step t. Note that time step is analogous to parsing a sequence of words sequentially in this context. $W_z, W_r \in \mathbb{R}^{d \times n}, W_h \in \mathbb{R}^{n \times n}$ are parameters of the GRU layer.

Tweet-level Attention The output of each GRU is a sequence of hidden vectors $h_1, h_2 \cdots h_L \in \mathbf{H}$, where $\mathbf{H} \in \mathbb{R}^{L \times n}$. Each hidden vector is n dimensions, which corresponds to the parameter size of the GRU. To learn a single n dimensional vector, the last hidden vector h_L is typically considered. However, a variety of pooling functions such as the average pooling, max pooling or attentional pooling can be adopted to learn more informative representations. More specifically, neural attention mechanisms are applied across the matrix $\mathbf{H}$, learning a weighted representation of all hidden vectors. Intuitively, this learns to select more informative words to be passed to subsequent layers, potentially reducing noise and improving model performance.

$$\mathbf{Y} = \tanh(W_y\, \mathbf{H}) \;\; ; \;\; a = \text{softmax}(w^\top\, \mathbf{Y}) \;\; ; \;\; r = \mathbf{H}\, a^\top$$

where $W_y \in \mathbb{R}^{n \times n}, w \in \mathbb{R}^n$ are the parameters of the attention pooling layer. The output $r \in \mathbb{R}^n$ is the final vector representation of the tweet. Note that the parameters of the attentional pooling layer are shared across all tweets and across both users.

Learning User Representations

Recall that each user is represented by K tweets and for each tweet we have a n dimensional vector. Let $t^i_1, t^i_2 \cdots t^i_K$ be all the tweets for a given user i. In order to learn a fixed n dimensional vector for each user, we require a pooling function across each user's tweet embeddings. In order to do so, we use a Coupled Attention Layer that learns to attend to U1 based on U2 (and vice versa). Similarly, for the negative sample, coupled attention is applied to (U1, U3) instead. However, we only describe the operation of (U1, U2) for the sake of brevity.

Coupled Attention The key intuition behind the coupled attention layer is to learn attentional representations of U1 with respect to U2 (and vice versa). Intuitively, this compares each tweet of U1 with each tweet of U2 and learns to weight each tweet based on this grid-wise comparison scheme. Let U1 and U2 be represented by a sequence of K tweets (each of which is a n dimensional vector) and let $T_1, T_2 \in \mathbb{R}^{k \times n}$ be the tweet matrix for U1 and U2 respectively. For each tweet pair (t^1_i, t^2_j), we utilize a feed-forward neural network to learn a similarity score between each tweet. As such, each value of the similarity grid is computed:

$$s_{ij} = W_c\, [t^1_i; t^2_j] + b_c \qquad (1)$$

where $W_c \in \mathbb{R}^{n \times 1}$ and $b_c \in \mathbb{R}^1$ are parameters of the feed-forward neural network. Note that these parameters are shared across all tweet pair comparisons. The score s_{ij} is a scalar value indicating the similarity between tweet i of U1 and tweet j of U2.

Aggregating Strong Signals Given the similarity matrix $\mathbf{S} \in \mathbb{R}^{K \times K}$, the strongest signals across each dimension are aggregated using max pooling. For example, by taking a max over the columns of $\mathbf{S}$, we regard the importance of tweet i of U1 as the strongest influence it has over all tweets of U2. The result of this aggregation is two K length vectors which are used to attend over the original sequence of tweets. The following operations describe the aggregation functions:

$$a^{row} = \text{smax}(\max_{row} \mathbf{S}) \quad \text{and} \quad a^{col} = \text{smax}(\max_{col} \mathbf{S}) \qquad (2)$$

where $a^{row}, a^{col} \in \mathbb{R}^K$ and smax is the softmax function. Subsequently, both of these vectors are used to attentively pool the tweet vectors of each user.

$$u_1 = T_1 \, a^{col} \quad \text{and} \quad u_2 = T_2 \, a^{row}$$

where $u_1, u_2 \in \mathbb{R}^n$ are the final user representations for U1 and U2.

Learning to Rank and Training Procedure

Given embeddings u_1, u_2, u_3, we introduce our similarity modeling layer and learning to rank objective. Given u_1 and u_2, the similarity between each user pair is modeled as follows:

$$s(u_1, u_2) = \frac{u_i \cdot u_2}{|u_1||u_2|} \qquad (3)$$

which is the cosine similarity function. Subsequently, the pairwise ranking loss is optimized. We use the margin-based hinge loss to optimize our model.

$$J = \max\{0, \lambda - s(u_1, u_2) + s(u_1, u_3)\} \qquad (4)$$

where λ is the margin hyperparameter, $s(u_1, u_2)$ is the similarity score for the ground truth (true couples) and $s(u_1, u_3)$ is the similarity score for the negative sample. This function aims to discriminate between couples and non-couples by increasing the margin between the ranking scores of these user pairs. Parameters of the network can be optimized efficiently with stochastic gradient descent (SGD).

Empirical Evaluation

Our experiments are designed to answer the following Research Questions (**RQ**s).

- **RQ1** - How well are machine learning and deep learning methods able to learn, predict, recommend relationships just based on linguistic information from social profiles? Are the romantic compatibility of two people predictable just based on textual information?
- **RQ2** - Does the amount of information (number of tweets per user) affect the ability to recommend relationships?
- **RQ3** - Are we able to derive any insight on how these models are learning to recommend relationships? Are attention models able to produce explainable relationship recommendations?

Experimental Setup

All empirical evaluation is conducted on our LoveBirds dataset which has been described earlier. This section describes the evaluation metrics used and evaluation procedure.

Evaluation Metrics Our problem is posed as a learning-to-rank problem. As such, the evaluation metrics used are as follows:

- **Hit Ratio @N** is the ratio of test samples which are correctly retrieved within the top N users. We evaluate on $N = 10, 5, 3$.
- **Accuracy** is the number of test samples that have been correctly ranked in the top position.
- **Mean Reciprocal Rank (MRR)** is a commonly used information retrieval metric. The reciprocal rank of a single test sample is the multiplicative inverse of the rank. The MRR is computed by $\frac{1}{Q} \sum_{i=1}^{|Q|} \frac{1}{rank_i}$.
- **Mean Rank** is the average rank of all test samples.

Evaluation Procedure Our experimental procedure samples 100 users per test sample and ranks the golden sample amongst the 100 negative samples.

Algorithms Compared In this section, we discuss the algorithms and baselines compared. Notably, there are no established benchmarks for this new problem. As such, we create 6 baselines to compare against our proposed CoupleNet.

- **RankSVM (Tf-idf)** - This model is a RankSVM (Support Vector Machine) trained on tf-idf vectors. This model is known to be a powerful vector space model (VSM) baseline. The feature vector of each user is a k dimensional vector, representing the top-k most common n-grams. The n-gram range is set to (1,3) and k is set to 5000 in our experiments. Following the original implementation, the kernel of RankSVM is a linear kernel.
- **RankSVM (Embed)** - This model is a RankSVM model trained on pretrained (static, un-tuned) word embeddings. For each user pair, the feature vector is the sum of all words of both users.
- **MLP (Embed)** - This is a Multi-layered Perceptron (MLP) model that learns to non-linearly project static word embedding. Each word embedding is projected using 2 layered MLP with ReLU activations. The user representation is the sum of all transformed word embeddings.
- **DeepCoNN (Deep Co-operative Neural Networks)** (Zheng, Noroozi, and Yu 2017) is a convolutional neural network (CNN). CNNs learn n-gram features by sliding weights across an input. In this model, all of a user's tweets are concatenated and encoded into a d dimensional vector via a convolutional encoder. We use a fixed filter width of 3. DeepCoNN was originally proposed for item recommendation task using reviews. In our context, we adapt the DeepCoNN for our RSR task (tweets are analogous to reviews). Given the different objectives (MSE vs ranking), we also switch[3] the factorization machine (FM) layer for the cosine similarity. The number of filters is 100. A max pooling layer is used to aggregate features.

[3] In our problem, we found that the FM layer significantly degraded performance.

- **Baseline Gated Recurrent Unit (GRU)** - We compare with a baseline GRU model. Similar to the DeepCoNN model, the baseline GRU considers a user to be a concatenation of all the user's tweets. The size of the recurrent cell is 100 dimensions.
- **Hierarchical GRU (H-GRU)** - This model learns user representations by first encoding each tweet with a GRU encoder. The tweet embedding is the last hidden state of the GRU. Subsequently, all tweet embeddings are summed. This model serves as an ablation baseline of our model, i.e., removing all attentional pooling functions.

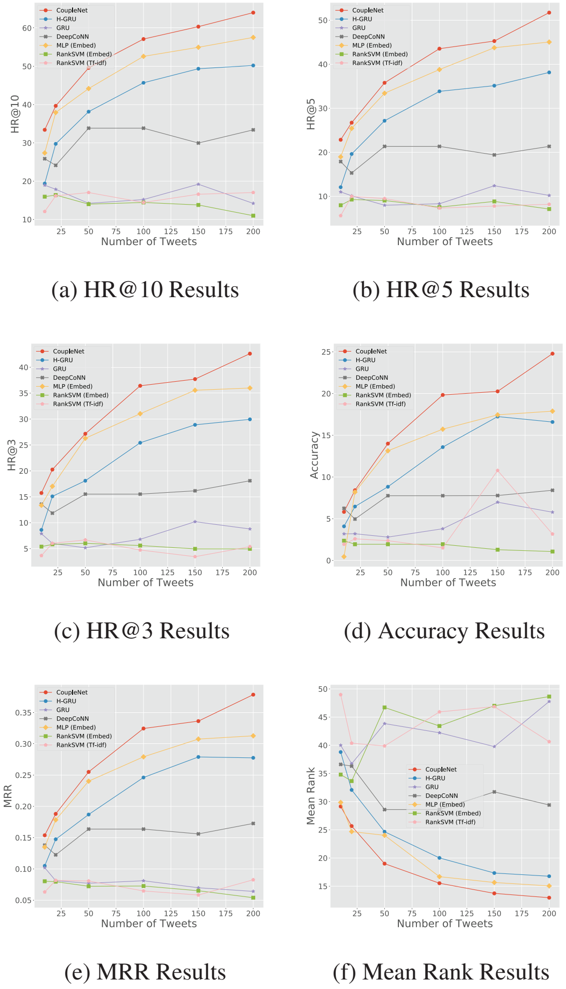
Implementation Details All models were implemented in Tensorflow on a Linux machine. For all neural network models, we follow a *Siamese* architecture (shared parameters for both users) and mainly vary the neural encoder. The cosine ranking function and hinge loss are then used to optimize all models. We train all models with the Adam (Kingma and Ba 2014) optimizer with a learning rate of 10^{-3} since this learning rate consistently produced the best results across all models. The batch size is tuned amongst $\{16, 32, 64\}$ and models are trained for 10 epochs. We report the result based on the best performance on the development set. The margin is tuned amongst $\{0.1, 0.2, 0.5\}$. All model parameters are initialized with Gaussian distributions with a mean of 0 and standard deviation of 0.1. The L2 regularization is set to 10^{-8}. We use a dropout of 0.5 after the convolution or recurrent layers. A dropout of 0.8 is set after the Coupled Attention layer in our model. Text is tokenized with NLTK's tweet tokenizer. We initialize the word embedding matrix with Glove (Pennington, Socher, and Manning 2014) trained on Twitter corpus. All words that do not appear more than 5 times are assigned unknown tokens. All tweets are truncated at a fixed length of 10 tokens. Early experiments found that raising the number of tokens per tweet does not improve the performance. The number of tweets per user is tuned amongst $\{10, 20, 50, 100, 150, 200\}$ and reported in our experimental results.

Discussion and Analysis

Figure 3 reports the experimental results on the LoveBirds2M dataset. For all baselines and evaluation metrics, we compare across different settings of η, the number of tweets per user that is used to train the model.

Firstly, we observe that CoupleNet significantly outperforms most of the baselines. Across most metrics, there is almost a $180\% - 200\%$ relative improvement over DeepCoNN, the state-of-the-art model for item recommendation with text data. The performance improvement over the baseline GRU model is also extremely large, i.e., with a relative improvement of approximately 4 times across all metrics. This shows that concatenating all of a user's tweets into a single document severely hurts performance. We believe that this is due to the inability of recurrent models to handle long sequences. Moreover, the DeepCoNN performs about 2 times better than the baseline GRU model.

On the other hand, we observe that H-GRU significantly improves the baseline GRU model. In the H-GRU model, sequences are only $L = 10$ long but are encoded K times with shared parameters. On the other hand, the GRU model has to process $K \times L$ words, which inevitably causes performance to drop significantly. While the performance of the H-GRU model is reasonable, it is still significantly outperformed by our CoupleNet. We believe this is due to the incorporation of the attentional pooling layers in our model, which allows it to eliminate noise and focus on the important keywords.



(a) HR@10 Results

(b) HR@5 Results

(c) HR@3 Results

(d) Accuracy Results

(e) MRR Results

(f) Mean Rank Results

Figure 3: Experimental Results on the LoveBirds2M dataset. Results are plotted against number of tweets. *Best viewed in color*. CoupleNet (*red*) outperforms all baselines.

A surprising and notable strong baseline is the MLP (Embed) model which outperforms DeepCoNN but still performs much worse than CoupleNet. On the other hand, RankSVM (Embed) performs poorly. We believe that this is attributed to the insufficiency of the linear kernel of the SVM. Since RankSVM and MLP are trained on the same features, we believe that nonlinear ReLU transformations of the MLP improve the performance significantly. Moreover, the MLP model has 2 layers, which learn different levels of abstractions. Finally, the performance of RankSVM (Tf-idf) is also poor. However, we observe that RankSVM (Tf-idf) slightly outperforms RankSVM (Embed) occasionally. While other models display a clear trend in performance with respect to the number of tweets, the performance of RankSVM (Tf-idf) and RankSVM (Embed) seem to fluctuate across the number of user tweets.

Finally, we observe a clear trend in performance gain with respect to the number of user tweets. This is intuitive because more tweets provide the model with greater insight into the user's interest and personality, allowing a better match to be made. The improvement seems to follow a logarithmic scale which suggests diminishing returns beyond a certain number of tweets. Finally, we report the time cost of COUPLENET. With 200 tweets per user, the cost of training is approximately $\approx$ 2 mins per epoch on a medium grade GPU. This is much faster than expected because GRUs benefit from parallism as they can process multiple tweets simultaneously.

Ablation Study

In this section, we study the component-wise effectiveness of COUPLENET. We removed layers from COUPLENET in order to empirically motivate the design of each component. Firstly, we switched CoupleNet to a pointwise classification model, minimizing a cross entropy loss. We found that this halves the performance. As such, we observe the importance of pairwise ranking. Secondly, we swapped cosine similarity for a MLP layer with scalar sigmoid activation (to ensure inputs lie within $[0, 1]$). We also found that the performance drops significantly. Finally, we also observe that the attention layers of COUPLENET contribute substantially to the performance of the model. More specifically, removing both the GRU attention and coupled attention layers cause performance to drop by 13.9%. Removing the couple attention suffers a performance degrade of 2.5% while removing the GRU attention drops performance by 3.9%. It also seems that dropping both degrades performance more than expected (not a straightforward summation of performance degradation).

Overall Quantitative Findings

In this subsection, we describe the overall findings of our quantitative experiments.

- Overall, the best HR@10 score for COUPLENET is about 64%, i.e., if an application would to recommend the top 10 prospective partners to a user, then the ground truth will appear in this list 64% of the time. Moreover, the accuracy is 25% (ranking out of 100 candidates) which is also reasonably high. Given the intrinsic difficulty of the problem, we believe that the performance of COUPLENET on this new problem is encouraging and promising. To answer **RQ1**, we believe that text-based deep learning systems for relationship recommendation are plausible. However, special care has to be taken, i.e., model selection matters.
- The performance significantly improves when we include more tweets per user. This answers **RQ2**. This is intuitive since more tweets would enable better and more informative user representations, leading to a better matching performance.

Model	HR@10
COUPLENET	64.1
w/o couple attention	61.6 (-2.5%)
w/o GRU attention	60.2 (-3.9%)
w/o GRU attention and couple attention	50.2 (-13.9%)
w/o cosine similarity	33.8 (-30.3%)
w/o pairwise (using pointwise)	36.1 (-28.0%)

Table 1: Component-wise ablation study with $\eta = 200$.

Qualitative Analysis

In this section, we describe several insights and observations based on real[4] examples from our LoveBirds20 dataset. One key advantage of COUPLENET is a greater extent of explainability due to the coupled attention mechanism. More specifically, we are able to obtain which of each user's tweets contributed the most to the user representation and the overall prediction. By analyzing the attention output of user pairs, we are able to derive qualitative insights. As an overall conclusion to answer **RQ3** (which will be elaborated by in the subsequent subsections), we found that COUPLENET is capable of explainable recommendations if there are explicit matching signals such as user interest and demographic similarity between user pairs. Finally, we discuss some caveats and limitations of our approach.

Mutual Interest between Couples is Captured in COUPLENET

We observed the COUPLENET is able to capture the mutual interest between couples. Table 2 shows an example from the LoveBirds2M dataset. In general, we found that most user pairs have noisy tweets. However, we also observed that whenever couple pairs have mutual interest, COUPLENET is able to assign a high attention weight to the relevant tweets. For example, in Table 2, both couples are fans of BTS[5], a Korean pop idol group. As such, tweets related to BTS are surfaced to the top via coupled attention. In the first tweet of User 1, tweets related to two entities, *seokjin* and *hoseok*, are ranked high (both entities are members of the pop idol group). This ascertains that COUPLENET is able to, to some extent, explain why two users are matched. This also validates the usage of our coupled attention mechanism. For instance, we could infer that User1 and User2 are matched because of their mutual interest in BTS. A limitation is that it is difficult to interpret why the other tweets (such as a *thank* you without much context, or *supporting your family*) were ranked highly.

[4] We do not explicitly report the actual user accounts in this paper because this might violate their privacy. Actual tweets are slightly modified to protect identities from search.

[5] https://en.wikipedia.org/wiki/BTS_(band)

Rank	User A	User B
1	i apologize to seokjin and hoseok	that's meant to say bts but imma too tired to
2	thank you!	more sorry for making such a mess
3	bts memes mayo	i'm not sure if I shld post this
4	@user @user support your family! Ł	the last couple of days have been shitty for me
5	welcome hun paramore!	blur pic effects are the best Ł

Table 2: Example of top-ranked tweets from user pair (ground truth is 1) in which mutual interests have the highest attention weight. Interest specific keywords are highlighted in red. CoupleNet successfully ranks this pair at the top position.

Rank	User C	User D
1	homecoming!	high school reception was a blast
2	taking meds for sports	preview will be out soon
3	so pumped for senior homecoming	this is my life homie

Table 3: Example of top-ranked tweets from user pair (ground truth is 1) which are ranked by the Coupled Attention layer. CoupleNet places school related tweets on the top.

CoupleNet Infers User Attribute and Demographic by Word Usage

We also discovered that CoupleNet learns to match users with similar attributes and demographics. For example, high school students will be recommended high school students at a higher probability. Note that location, age or any other information is not provided to CoupleNet. In other words, user attribute and demographic are solely inferred via a user's tweets. In Table 3, we report an example in which the top-ranked tweets (via coupled attention) are high school related tweets (homecoming, high school reception). This shows two things: (1) the coupled attention shows that the following 3 tweets were the most important tweets for prediction and (2) CoupleNet learns to infer user attribute and demographic without being explicitly provided with such information. We also note that both users seem to have strongly positive tweets being ranked highly in their attention scores which might hint at the role of sentiment and mood in making prediction.

CoupleNet Ranks Successfully Even Without Explicit Signals

It is intuitive that not every user will post interest or demographic revealing tweets. For instance, some users might exclusively post about their emotions. When analyzing the ranking outputs of CoupleNet, we found that, interestingly, CoupleNet can successfully rank couple pairs even when there seem to be no explicit matching signal in the social profiles of both users.

Rank	User E	User F
1	wanna be treated like a princess Ł	can't deal with this forever
2	in bed with cosy clothes and fluffy socks	my diet is screwed
3	rt if you are currently in a mess	feel too sick
4	so much regret lmao	life is shit, home is shit
5	some girls are just so naturally pretty	still care about my grades

Table 4: Example of top-ranked tweets (from attention) from user pair (ground truth is 1) in which there is no explicit signal. CoupleNet correctly ranks this user pair at top position.

Table 4 shows an example where two user profiles do not share any explicit matching signals. User E and User F are a ground truth couple pair and the prediction of CoupleNet ranks User E with User F at the top position. The top tweets of User E and User F are mostly emotional tweets that are non-matching. Through this case, we understand that CoupleNet does not simply match people with similar emotions together. Notably, relationship recommendation is also a problem that humans may struggle with. Many times, the reason why two people are in a relationship may be implicit or unclear (even to humans). As such, the fact that CoupleNet ranks couple pairs correctly even when there is no explicit matching signals hints at its ability to go beyond simple keyword matching. In this case, we believe 'hidden' (latent) patterns (such as emotions and personality) of the users are being learned and modeled in order to make recommendations. This shows that CoupleNet is not simply acting as a text-matching algorithm and learning features beyond that.

Side Note, Caveats and Limitations

While we show that our approach is capable of producing interpretable results (especially when explicit signals exist), the usefulness of its explainability may still have limitations, e.g., consider Table 4 where it is clear that the results are not explainable. Firstly, there might be a complete absence of any interpretable content in two user's profiles in the first place. Secondly, explaining relationships are also challenging for humans. As such, we recommend that the outputs of CoupleNet to be only used as a reference. Given that a user's profile may contain easily a hundreds to thousands of tweets, one posssible use is to use this ranked list to enable more efficient analysis by humans (such as social scientist or linguists). We believe our work provides a starting point of explainable relationship recommendation.

Conclusion

We introduced a new problem of relationship recommendation. In order to construct a dataset, we employ a novel distant supervision scheme to obtain real world couples from social media. We proposed the first deep learning model for text-based relationship recommendation. Our deep learning model, COUPLENET is characterized by its usage of hierarchical attention-based GRUs and coupled attention layers. Performance evaluation is overall optimistic and promising. Despite huge class imbalance, our approach is able to recommend at a reasonable precision (64% at HR@10 and 25% accuracy while being ranked against 100 negative samples). Finally, our qualitative analysis shows three key findings: (1) COUPLENET finds mutual interests between users for match-making, (2) COUPLENET infers user attributes and demographics in order to make recommendations, and (3) COUPLENET can successfully match-make couples even when there is no explicit matching signals in their social profiles, possibly leveraging emotion and personality based latent features for prediction.

References

Akehurst, J.; Koprinska, I.; Yacef, K.; Pizzato, L. A. S.; Kay, J.; and Rej, T. 2011. CCR - A content-collaborative reciprocal recommender for online dating. In *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence*.

Arnoux, P.-H.; Xu, A.; Boyette, N.; Mahmud, J.; Akkiraju, R.; and Sinha, V. 2017. 25 tweets to know you: A new model to predict personality with social media.

Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Benton, A.; Arora, R.; and Dredze, M. 2016. Learning multiview embeddings of twitter users. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016*.

Cho, K.; van Merrienboer, B.; Gülçehrse, Ç.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR* abs/1406.1078.

Cobb, C., and Kohno, T. How public is my private life?: Privacy in online dating. In *Proceedings of the 26th International Conference on World Wide Web,WWW 2017*.

Diaz, F.; Metzler, D.; and Amer-Yahia, S. Relevance and ranking in online dating systems. In *Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2010*.

Gupta, P.; Goel, A.; Lin, J.; Sharma, A.; Wang, D.; and Zadeh, R. 2013. Wtf: The who to follow service at twitter. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, 505–514. New York, NY, USA: ACM.

Hancock, J. T.; Toma, C.; and Ellison, N. 2007. The truth about lying in online dating profiles. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, 449–452. ACM.

He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X.; and Chua, T.-S. 2017. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17.

Hu, Y.; Koren, Y.; and Volinsky, C. 2008. Collaborative filtering for implicit feedback datasets. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, 263–272. Ieee.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *CoRR* abs/1412.6980.

Krzywicki, A.; Wobcke, W.; Kim, Y. S.; Cai, X.; Bain, M.; Compton, P.; and Mahidadia, A. 2014. Evaluation and deployment of a people-to-people recommender in online dating.

Luong, M.-T.; Pham, H.; and Manning, C. D. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Maldeniya, D.; Varghese, A.; Stuart, T.; and Romero, D. 2017. The role of optimal distinctiveness and homophily in online dating.

Masden, C., and Edwards, W. K. 2015. Understanding the role of community in online dating. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, 535–544. New York, NY, USA: ACM.

Nagarajan, M., and Hearst, M. A. 2009. An examination of language use in online dating profiles. In *Proceedings of the Third International Conference on Weblogs and Social Media, ICWSM 2009, San Jose, California, USA, May 17-20, 2009*.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP*.

Severyn, A., and Moschitti, A. 2015. Learning to rank short text pairs with convolutional deep neural networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Tay, Y.; Phan, M. C.; Luu, A. T.; and Hui, S. C. 2017. Learning to rank question answer pairs with holographic dual LSTM architecture. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2017*.

Tay, Y.; Luu, A. T.; and Hui, S. C. 2017. Translational recommender networks. *arXiv preprint arXiv:1707.05176*.

Tay, Y.; Tuan, L. A.; and Hui, S. C. 2018. Multi-pointer co-attention networks for recommendation. *CoRR* abs/1801.09251.

Toma, C. L., and Hancock, J. T. 2010. Reading between the lines: Linguistic cues to deception in online dating profiles. In *Proceedings of the CSCW, 2010*.

Tu, K.; Ribeiro, B.; Jensen, D.; Towsley, D.; Liu, B.; Jiang, H.; and Wang, X. 2014. Online dating recommendations: Matching markets and learning preferences. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14 Companion, 787–792. New York, NY, USA: ACM.

Wei, H.; Zhang, F.; Yuan, N. J.; Cao, C.; Fu, H.; Xie, X.; Rui, Y.; and Ma, W.-Y. 2017. Beyond the words: Predicting user personality from heterogeneous information. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, WSDM '17, 305–314. New York, NY, USA: ACM.

Xia, P.; Jiang, H.; Wang, X.; Chen, C.; and Liu, B. 2014. Predicting user replying behavior on a large online dating site.

Xia, P.; Liu, B.; Sun, Y.; and Chen, C. 2015. Reciprocal recommendation system for online dating. In *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*.

Yang, Z.; Yang, D.; Dyer, C.; He, X.; Smola, A. J.; and Hovy, E. H. 2016. Hierarchical attention networks for document classification.

Zheng, L.; Noroozi, V.; and Yu, P. S. 2017. Joint deep modeling of users and items using reviews for recommendation. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, 425–434. ACM.