THE LOVE EQUATION: COMPUTATIONAL MODELING OF ROMANTIC RELATIONSHIPS IN FRENCH DRAMA

A study by: Folgert Karsdorp, Mike Kestemont, Christof Schöch, and Antal van den Bosch
Given a play and a speaker, can we identify that speaker’s lover?
FRENCH DRAMA

- Théâtre Classique
  - French plays from the 17\textsuperscript{th} & 18\textsuperscript{th} centuries
  - Various genres: Comedies, tragedies, etc.
- Speakers, not characters
WHAT IS LOVE?

- Romantic interest
- One single lover
- Reciprocal vs. unidirectional
APPROACH

- Matchmaking problem
- Ranking problem
- Define features and use machine learning
APPROACH: RANKING

- Given a query speaker, we refer to all other speakers as candidates
- Exactly one candidate is the query speaker’s lover
- We want to rank the candidates such that query speaker’s lover has the highest ranking
THE DATA

- 720 plays
  - 9.3 million word tokens
  - 6500 speakers
- Encoded in XML
  - Also includes metadata about the settings, speakers, etc.
THE DATA
THE DATA

- 295 love relationships
  - 90 of which are reciprocal
- 2 Datasets:
  1. Reciprocal relationships only
  2. All relationships
THE MODEL: PEGASOS

- Linear fit
- Application of the stochastic gradient descent method
- Since it runs randomly over the samples, the evaluation was done on the average of 10 runs with different random seeds
THE MODEL: PEGASOS

INPUT: $S$, $\lambda$, $T$
INITIALIZE: Set $w_1 = 0$
FOR $t = 1, 2, \ldots, T$
    Choose $i_t \in \{1, \ldots, |S|\}$ uniformly at random.
    Set $\eta_t = \frac{1}{\lambda t}$
    If $y_{i_t} \langle w_t, x_{i_t} \rangle < 1$, then:
        Set $w_{t+1} \leftarrow (1 - \eta_t \lambda)w_t + \eta_t y_{i_t} x_{i_t}$
    Else (if $y_{i_t} \langle w_t, x_{i_t} \rangle \geq 1$):
        Set $w_{t+1} \leftarrow (1 - \eta_t \lambda)w_t$
[ Optional: $w_{t+1} \leftarrow \min \left\{ 1, \frac{1/\sqrt{\lambda}}{\|w_{t+1}\|} \right\} w_{t+1} \]  
OUTPUT: $w_{T+1}$
EVALUATION: LEAVE ONE OUT

- Completely ignore a speaker $s_q$ when training the algorithm
- Include $s_q$ (and his/her relationships) in the test data
- Use the trained algorithm to generate rankings for $s_q$
EVALUATION: MRR

- The reciprocal rank of one query speaker is the inverse of the ranking of his/her lover
  - 1 if the lover is in the 1st place, 0.5 if he/she is in the 2nd place, etc.
- The mean reciprocal rank (MRR) is, as the name implies, the mean of the reciprocal ranks of all queries
FEATURES: OUTLINE

- Speaker similarity
- Analogous lovers
- Word similarity
- Word analogy
- Interaction frequency
- Scene co-occurrence
- Gender
SPEAKER VECTORS

- Inspired by the Pragraph Vector model
- Which is…?
THE PARAGRAPH VECTOR MODEL

- Inspired by vector representation of words
- And that would be...?
VECTOR REPRESENTATION OF WORDS

- Used as a method to predict the next word given the words so far
- Each word in the text is represented as a vector
- Word tokens share these vectors
- Given some words in the text, combine their vector representation to get a representation of the predicted next word
VECTOR REPRESENTATION OF WORDS

Classifier

Average/Concatenate

Word Matrix

the

cat

sat
THE PARAGRAPH VECTOR MODEL

- Each paragraph in the text is represented by a unique vector
- When making the prediction, combine the word vectors and the paragraph vector together
- It can be thought of as a memory that remembers what is missing from the current context
THE PARAGRAPH VECTOR MODEL
THE PARAGRAPH VECTOR MODEL

- Unsupervised learning
- Initialize the paragraph and word vectors randomly
- Use stochastic gradient descent to learn the vectors until convergence is achieved
SPEAKER VECTORS

- Like in the Pragraph Vector model, we would like to do a mapping to unique vectors
- This time, we will map the speakers!
- Practically speaking, we just look at all of the verses of a speaker as a single paragraph
- We can use the speaker vectors to compare two speakers’ distributional semantics
FEATURE 1: SPEAKER SIMILARITY

• We expect lovers to speak about similar topics in the same way

• Given a query speaker with speaker vector $s_q$ and a candidate with speaker vector $s$, we compute the cosine similarity of $s_q$ and $s$
FEATURE 1: SPEAKER SIMILARITY

- In Pierre Corneille’s Le Menteur, Alcippe and Clarice are lovers
- Reducing the speaker vectors to a 2 dimensional space gives a good clue about how similar they are
- Alcippe’s main contestant is close by
FEATURE 2: ANALOGOUS LOVERS

• We expect the relation between lovers to be similar to the relation between other lover pairs
• Given a query speaker with speaker vector $s_q$ and a candidate with speaker vector $s$, we compute the displacement vector $D = s_q - s$
• We compute the cosine similarity between $D$ and the displacement vectors of all known lover pairs
• The feature value is the maximum similarity found
FEATURE 2: ANALOGOUS LOVERS

- In Pierre Corneille’s Clitandre, of all relations between Rosidor and any other speaker, the one with Caliste is the most similar to the relation between Alcippe and Clatice
FEATURE 3: WORD SIMILARITY

• We expect everyone to talk about lovers in the same way

• Given a query speaker with word vector $s_q$ and a candidate with word vector $s$, we compute the cosine similarity of $s_q$ and $s$

• We replace first person pronouns with the name of the speaker
FEATURE 4: WORD ANALOGY

• Given a query speaker with \textit{word} vector $s_q$ and a candidate with \textit{word} vector $s$, we compute the maximal cosine similarity between the displacement vector $D = s_q - s$ and the displacement vectors of all known lover pairs.
FEATURE 5: INTERACTION FREQUENCY

- We expect lovers to interact with each other more frequently than non-lovers.
- For each speaker we calculate the normalized count of how often he/she interacts with all other speakers.
- Can be viewed as a weighted graph.
**FEATURE 5: INTERACTION FREQUENCY**

- In Pierre Corneille’s *La Suivante*, Daphnis is Florame’s lover.
- Looking at Florame’s interactions, we see that he most frequently interacts with Daphnis.
- Phenomenon: Florame also frequently interacts with Theante, who is his main contestant.
FEATURE 6: SCENE CO-OCCURRENCE

- We expect lovers to share more scenes with each other than non-lovers.
- Same as F5, but with respect to appearing in the same scene rather than interacting with each other.
FEATURE 7: GENDER

• We expect lovers to be of different genders

• Given a query speaker $s_q$ and a candidate $s$, we mark the feature value as 1 if $s_q$ and $s$ are of different genders, and 0 if they’re of the same gender
RESULTS

- Reciprocal dataset
  - MRR: 0.9
  - Accuracy: 0.81
- Full dataset
  - MRR: 0.87
  - 0.75
- Pretty impressive!
RESULTS: EXPERIMENTATION

- What if the features interact with each other in non-linear ways?
- How will a certain feature do on its own?
- How will the model do without a certain feature?
RESULTS: EXPERIMENTATION

- When only a single feature was introduced to the model, the word similarity feature gave the best performance.
- Gender gave the worst performance.
RESULTS: EXPERIMENTATION

- When a feature was excluded from the model, the gender feature had the greatest impact.
- Excluding gender gave worse results than including word similarity on its own.
# Results: Experimentation

<table>
<thead>
<tr>
<th>Feature</th>
<th>With Feature</th>
<th>Without Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Reciprocal</td>
<td>Full</td>
</tr>
<tr>
<td></td>
<td>MRR @1</td>
<td>MRR @1</td>
</tr>
<tr>
<td>F1. Speaker Similarity</td>
<td>0.51 0.29</td>
<td>0.51 0.28</td>
</tr>
<tr>
<td>F2. Analogous Lovers</td>
<td>0.41 0.18</td>
<td>0.48 0.27</td>
</tr>
<tr>
<td>F3. Word Similarity</td>
<td><strong>0.74 0.59</strong></td>
<td><strong>0.73 0.56</strong></td>
</tr>
<tr>
<td>F4. Word Analogy</td>
<td>0.45 0.24</td>
<td>0.41 0.22</td>
</tr>
<tr>
<td>F5. Interaction Frequency</td>
<td>0.53 0.28</td>
<td>0.55 0.32</td>
</tr>
<tr>
<td>F6. Scene Co-occurrence</td>
<td>0.53 0.32</td>
<td>0.51 0.28</td>
</tr>
<tr>
<td>F7. Gender</td>
<td>0.29 0.07</td>
<td>0.37 0.12</td>
</tr>
<tr>
<td>F1. – F7.</td>
<td><strong>0.9 0.81</strong></td>
<td><strong>0.87 0.75</strong></td>
</tr>
</tbody>
</table>
• Our query speaker is Suzanne from Beaunoir’s Le Sculpteur
• Her lover is Le Doux
FUTURE WORK

- Investigate harmful interactions between features
- Perfect the speaker vector representation
- Generalize the problem to a multiple lover question
- Different types of relationships
- Different types of texts
QUESTIONS?
HONORARY MENTION: MARVEL UNIVERSE AS A SOCIAL NETWORK

- Analysis of the Marvel Comics Universe as a social network yielded some interesting results
- For example, a distribution function for the number of partners of a character could be identified
HONORARY MENTION: MARVEL UNIVERSE AS A SOCIAL NETWORK

\[ P(k) \sim k^{-0.7158} 10^{-k/2167} \]
RESOURCES

- https://pure.knaw.nl/portal/files/1057319/paper4_2.pdf
- https://github.com/dramacode/moliere
- https://en.wikipedia.org/wiki/Mean_reciprocal_rank