Institute of Cognitive Science

Bachelor's Thesis

A textual recommender system and other text mining applications for clinical data

Philipp Hummel

May 12, 2017

First supervisor: Dr. Frank Jäkel
Second supervisor: Prof. Dr. Gordon Pipa

A textual recommender system and other text mining applications for clinical data

In unusual clinical situations doctors would often like to review cases of similar patients to guide their decision making for the current one. To retrieve the relevant cases however is a hard and time consuming task. This thesis works on building a recommender system that automatically finds physician letters similar to a reference letter in an information retrieval like manner. We use a small dataset of free text physician letters of oncology patients to do exploratory work on this issue. We provide a prototypical system that gives recommendations on this dataset and verify its performance through a psychological experiment assessing doctor's similarity judgments of those letters. We find that the similarity measure of our recommender system correlates strongly with doctor's similarity judgments. Additionally we explore other text mining applications like automatic diagnosis extraction based on this dataset.

Acknowledgements

Contents

Chapter 1

Introduction

1.1 Motivation

Publishing the case of a patient with a particularly interesting medical phenomenon in the form of a clinical case report has seen a change in popularity in the medical community. The number of published case reports has been declining in standard journals. This has happened not only because case reports can hardly contribute to a good impact factor, but also because their scientific benefit has been questioned (Mason, 2001). However, several new journals dedicated only to case reports have emerged (Kidd and Hubbard, 2007)(how to cite a journal's existence, which does not have an introductory article?) and many people have argued for the value of case reports for research itself but also beyond (Williams, 2003; Dib et al., 2008; Sandu et al., 2016). Importantly case reports allow practitioners a more in-depth understanding of specific disease courses and provide educational material for students (Nissen and Wynn, 2014). Although case reports lack the scientific validity of large empirical studies, it is apparent that people have strong intuitions for the usefulness of them. During our collaboration with practicing doctors we also found that practitioners use the clinical records of similar patients to guide problem solving for the current patient. Especially when faced with hard or unusual cases doctors seek similar patient information from the hospital database. While this only shows that doctors think that the presentation of similar cases helps them in their work, we will argue that this can indeed improve their medical problem solving.

Cognitive Scientists have discussed the usefulness of examples for reasoning processes for long and found that at least in some experimental settings reasoning processes are based on earlier presented examples (Medin and Schaffer, 1978). More recently these reasoning processes have also been studied in more realistic scenarios. Klein (2008) reviews models for decision making under real world circumstances. According to him experts interpret a situation based on its resemblance to remembered situations. Once a sufficiently similar situation has been retrieved from memory, experts apply the solution from the remembered situation to the current one in a thought experiment. They evaluate whether or not this solution strategy will

lead to success and adjust it, if necessary. In case no way to adequately adjust the solution can be found, another situation is retrieved from memory. This process is repeated until a sufficiently good solution is found. Presentation of similar cases should therefore aid doctor's decision making in an actual clinical setting. The medical domain has also been directly addressed by research in cognitive science. Elstein and Schwarz (2002) have concluded that for medical problem solving reasoning processes can be divided into two distinct categories. For cases perceived as easy doctors apply a kind of pattern recognition based on the examples they have encountered before and use solutions stored in memory. For harder cases, however, doctors need to rely on a more elaborate reasoning process. They have to consciously generate and eliminate hypotheses to be able to solve the problem. It is plausible that hypothesis generation as well as hypothesis falsification is also guided by the doctor's experience of earlier patients. From a more theoretical perspective Kolodner and Kolodner (1987) have specifically argued that "[i]ndividual experiences act as exemplars upon which to base later decisions" in medical problem solving. Their research was partially driven by the desire to understand the way in which clinicians perform problem solving but also by the goal of building artificial systems that can aid in this process. They argue that both humans and machines can learn from specific examples and use them to reason about new problems.

Around the idea that artificial systems might learn from examples has evolved a whole branch of Artificial Intelligence (AI), which is called "Case-based reasoning". This domain has been greatly influenced by psychological findings, some of them mentioned above. Researchers have successfully built systems used in real world applications, that reason from the examples provided (Aamodt and Plaza, 1994). Within this domain of AI one of the greatest application areas is medicine. It seems that this area does not only offer straight forward usage of examples, but also has a need for automatic aids for problem solving (Begum et al., 2011).

Given the practical, psychological and theoretical reflections above we believe that it would be helpful for practitioners to be able to review cases of similar patients. One particularly well suited source for the retrieval of patient cases are databases of physician letters, as these letters provide concise summaries of the specifics of a patient that matter in practice. Search in these databases is, to our knowledge, usually limited to character matching procedures and therefore provides limited practical value for doctors. We therefore set out to build a prototypical recommender system on those physician letters to do automatic retrieval of only the relevant documents from a database.

1.2   Dataset

To get a dataset for exploratory work on the recommender system we collaborated with the university hospital Freiburg. This hospital has a database of approximately 190,000 German physician letters in PDF form (Spadaro, 2012) (in der clinicon Beschreibung heißt es 190.000 medizinische Dokumente und Arztbriefe) (noch nicht

zufrieden, wie das item in der bibliography erscheint. Ich weiß allerdings auch nicht wie es richtig wäre.). These physician letters are free text documents written by the doctors to keep record of the patient's visit. They usually include information about the patient's age, sex, diagnosed diseases, therapy history, current complaints, many more medical details like blood counts, but also personal information like names and birth dates. The letters generally follow a rough structure. Almost all of them include a letter head (a greeting and introduction), a diagnosis (summarizing diagnosed diseases bullet point like), a therapy history (listing the past therapies with dates) and an anamnesis (free text about current complaints etc.) section, separated into individual paragraphs. In principal though, doctors are free to document this information in the way they please. The database does not, however, contain the information of 190,000 unique patients. For many patients several letters are included, as a new visit will often result in an updated letter, that is added to the database. We refer to these letters as "follow-up" letters.

To get permission to use a subset of those letters for our experiment, it was necessary to ensure that all personal information was removed from them. A medical student of the university hospital was therefore paid to manually anonymize 307 of the letters and forward them to us. The letters were given to us in Microsoft Word XML format.

With this data at hand we build a prototypical recommender system, that retrieves similar cases based on one physician letter. To achieve this goal we first clear our dataset from duplicates with the help of basic information retrieval methods. This is described in the subsequent chapter. We elaborate on methods for hiding and showing specific information from the letters in chapter 3. For this goal we make extensive use of more advanced information retrieval methods and introduce them alongside there. Chapter 4 is concerned with classification of the physician letters based on qualities of interest of the corresponding patients. Chapter 5 describes the recommender system itself. Experimental assessments of the quality of the system are explained in chapter 6. Finally we review shortcomings, conceivable problems and possibilities for further work in the discussion.

Chapter 2

Dataset Cleansing

The dataset we acquired from the university hospital contains several duplicate letters. To ensure undistorted test results, we have to identify as many of them as possible. It is clear that finding all of these duplicates manually is not only error-prone, but also very time consuming as in principle $\frac{(n-1)^2+(n-1)}{2}$ letter comparison have to be made. With $n = 307$ this amounts to almost 50,000 comparisons. Additionally personal information useful for this task, like names and birthdates, has been removed during anonymization. Therefore we make use of two semi-automatic procedures for duplicate identification. First we use a longest common subsequence matching method. This method would be sufficient for our problems, if we only had to deal with exact duplicates. However, we face the issue of follow-up letters with modified information in our dataset. To overcome this problem we use a procedure well known in the information retrieval community: The bag of words model for representing texts as vectors, with which it is possible to compute distances between documents. Duplicates are then found by their vector proximity.

2.1 The Bag of Words Model

The bag of words model represents a text as the multiset (bag) of its words. That means all word order information is disregarded and only the information how often a word is present in a text is encoded (Manning et al., 2008a). (Mit der Art wie das Buch hier zitiert wird bin ich noch nicht glücklich. Es sollte eigentlich chapter 2 und 6 sein und nicht 2008 a und b.) More concretely in the bag of words model documents are represented as fixed length feature vectors, where every feature is a word occurrence count. In the simplest approach every feature is the word or term frequency $\mathrm{tf}(t, d)$ of term t in document d. Where $\mathrm{tf}(t, d)$ is the occurrence count of word or term t in document d. To compute feature vectors for documents in a corpus the vocabulary V of the corpus needs to be established first. Documents are represented by a $1 \times |V|$ vector of the values $\mathrm{tf}(t, d)$ for all $t \in V$. To represent text in the bag of words model, however, the text needs to be preprocessed first.

Preprocessing

In computers text is most often represented as sequences of characters. The process of extracting words from such a sequence is known as tokenization (Manning et al., 2008b). A first naive approach to tokenization might be to split the sequence of characters on every whitespace and regard everything in between as a word. This approach, however, can easily lead to unexpected results. Consider the example string "The doctor treats the patient.". The naive approach yields the words "The", "doctor", "treats", "the" and "patient.". Note how the last word contains the punctuation character ".". So even for very easy examples the naive approach does not produce expected results. Luckily many more sophisticated algorithms are implemented in ready-to-use open-source software packages.

Some words are more important or representative of a text than others and so it is a useful preprocessing step for the bag of words model to remove some frequently appearing but uninformative words, so called stop words. A list of English stop words comprises words like "the", "a", "just" or "some". While they are necessary for the grammatical structure of a language, they do not convey much information about the similarity of documents (Manning et al., 2008b).

Additionally for the bag of words model we are not interested in the exact grammatical form in which a word is present in a text. Removing this unnecessary information is achieved with a procedure called stemming, that maps word appearances to their stem. It maps words such as "doctors" and "doctor" both to their common stem "doctor". Both removal of stop words and stemming seem at first glance to disregard information. This is true, but they allow to represent a text more compactly and thereby reduce the noise of the representation. The following example might provide clarification about the procedures.

Bag of Words Example

Assume the corpus consists of two documents $d_1 =$"The patients with disease A." and $d_2 =$"The patients with disease B.". After tokenization, removal of stop words and stemming the vectors representing the two texts are:

$$
\mathbf{v}_{d_1} = \begin{matrix} 1 \\ 1 \\ 1 \\ 0 \end{matrix} \quad \mathbf{v}_{d_2} = \begin{matrix} 1 \\ 1 \\ 0 \\ 1 \end{matrix} \quad \begin{matrix} \text{patient} \\ \text{disease} \\ \text{A} \\ \text{B} \end{matrix}
$$

Note how the punctuation character does not appear, as the tokenization has removed it from the list of words. The words "the" and "with" have been removed as stop words and the word "patients" has been stemmed to produce "patient". Then the list of words has been converted to a bag of words vector.

Application and shortcomings

The vectors $\mathbf{v}_{d_1}$ and $\mathbf{v}_{d_2}$ live in a four dimensional vector space and we can either use them as feature vectors for a machine learning task or compute the distance between them to get a measure of dissimilarity. The standard distance measure used for bag of words vectors is the cosine distance $d_{\cos}(\mathbf{v}_1, \mathbf{v}_2) = \frac{\mathbf{v}_1 \cdot \mathbf{v}_2}{||\mathbf{v}_1|| \cdot ||\mathbf{v}_2||}$, where $\mathbf{v}_1 \cdot \mathbf{v}_2$ is the scalar product of $\mathbf{v}_1$ and $\mathbf{v}_2$. This distance measure is preferred over the standard euclidean distance $d_{\mathrm{euclid}}(\mathbf{v}_1, \mathbf{v}_2) = \sqrt{\mathbf{v}_1 \cdot \mathbf{v}_2}$, as it normalizes the scalar product, by the product of the lengths of the vectors. Thereby only the relative word frequencies are important and not the absolute ones. I.e. pairs of longer documents do not have a bigger distance, just because the absolute word counts are higher.

Generally the bag of words model has severe limitations. Consider for example the two sentences $d_3 =$"The patients with disease A, but not B" and $d_4=$"The patients with disease B, but not A". Their representation in the bag of words model is equivalent, although they express quite different meanings. Still the bag of words model is a standard approach for information retrieval problems, as it has many desirable properties such as the fixed length representation and straight forward embedding of texts.

2.2 Duplicate Detection

With the bag of words model and the longest common subsequence matching method we semi-automatically identify duplicates in our data. We had an expert identify all duplicates present in a subset of 150 of these letters manually to have a baseline to which we can compare the methods' performances.

The longest common subsequence matching procedure scans two documents and finds the longest sequence of characters they have in common. If the length of this sequence exceeds a threshold the pair is marked as possibly duplicate and manually inspected. We lower the threshold until the false positive rate becomes to high. Secondly we preprocess all texts as described above and map them into the bag of words representation. We use the cosine distance to compute a measure of dissimilarity between pairs and mark all pairs with distance lower than a threshold as possibly duplicate and again iteratively increase the threshold.

The bag of words approach identifies all manually detected duplicates and follow-ups while having a very low false positive rate. It detects one additional manually undetected follow-up pair. The string matching procedure is not as useful due to a high false positive rate and does not detect all manually found duplicates. Its performance is worse than the performance of the bag of words procedure, especially for follow-up letters. However, it finds a second manually undetected follow-up pair, that we have not identified with the bag of words procedure either. We use both approaches with thresholds from the "training" set on the remaining 157 letters as well. Overall we detect 18 exact copy pairs and 17 follow-up pairs. We additionally remove three letters from the dataset as they include almost no information (an artifact of the specific documentation procedure at the university hospital). This

results in our final dataset consisting of 269 unique physician letters (follow-ups are only included, where mentioned explicitly).

Chapter 3

Fine-grained information presentation

3.1 Paragraph Extraction

As already mentioned above, the letters almost always contain separate paragraphs like greeting, diagnosis, therapy history and anamnesis. To be able to hide unnecessary information or to only present requested information it is useful to automatically extract individual paragraphs from the documents. As the documents are similar in structure and the word XML format allows to automatically examine the XML tree structure of a document, we use a rule based approach for extracting the individual paragraphs. A simplified rule to find the beginning of the diagnosis paragraph is shown in pseudocode:

```
diagnosis_regex = '[dD]iagnose(n)?'
text = this_xml_node.text()
if diagnosis_regex.match(text)
        and boldface(text)
        and preceded_by_newline(this_xml_node):
then:
        diagnosis_start = this_xml_node
```

A regular expression is defined that matches the beginning of the diagnosis paragraph (the German word for diagnosis is "Diagnose"). The text of every node in the XML tree is checked for a regular expression match and several other rules. If an XML node matches all criteria it is marked as the beginning of the diagnosis paragraph. With a set of rules like the one above we automatically extract the paragraphs of interest from the documents. This approach, however, is not completely reliable, as the doctors are free to write the documents in the way they please. Indeed we find several wrongly extracted paragraphs, that e.g. include the subsequent paragraph as well. For our dataset it is possible to check the extraction process by hand.

However, this is tedious work and is not scalable to bigger datasets. We therefore
explore whether we can in principle make use of other automated methods to find
paragraphs for which the extraction process does not produce desired results. We
therefore take several correctly extracted and one incorrectly extracted diagnosis
paragraphs and convert them to their bag of words representation. To get a feeling
for how these vectors behave we use Principle Component Analysis (PCA) to get a
lower dimensional approximation of the vectors. PCA finds the linear subspace with
desired dimensionality of the original space that preserves as much of the variance of
the vectors in the original space as possible. Thereby one can gain a low dimensional
approximation of the high dimensional data and use this approximation for visual
inspection. See figure 3.1 for a 2D PCA plot of the bag of words representation
of the correctly and incorrectly extracted diagnose paragraph. As is apparent from
the figure, it would not be a hard task to automatically detect the outlier. In this
case the incorrectly extracted paragraph includes not only the diagnosis, but also
the therapy history. In cases like this with additional text present, it is an easy
task to identify the incorrect ones. A harder problem arises, when only parts of the
paragraph of interest have been extracted. However, we believe that this problem
is of little concern. The way our rules are built it is very unlikely that we will face
this problem. The paragraph would have to include an empty line, the subsequent
one would have to contain only boldface characters and a few more conditions would
have to be fulfilled for this problem to arise. Indeed, we did not find a case of this
problem in our dataset. (Ich könnte auch mal tatsächliche outlier detection machen,
wenn du das für sinnvoll hälst. Hab ein paar Ideen, die gut funktionieren könnten,
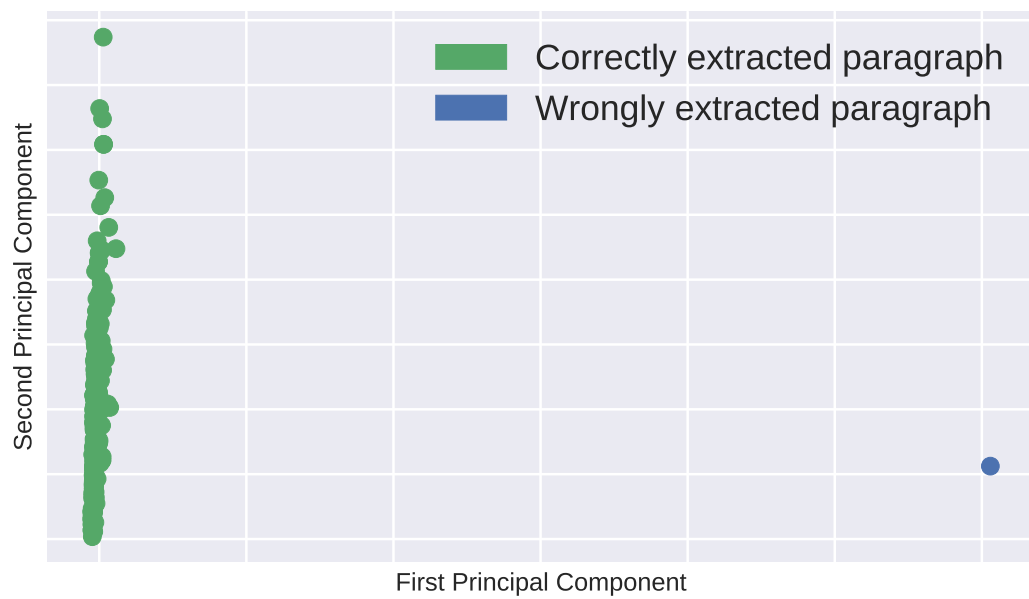wollte aber keine Zeit rein stecken, falls wir es nicht benutzen wollen.)

Figure 3.1: 2D PCA projection of bag of words representation of one incorrectly extracted diagnosis paragraph and several correctly extracted ones.

Bibliography

Aamodt, A. and Plaza, E. (1994), 'Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches', *AI COMMUNICATIONS* **7**(1), 39–59.

Begum, S., Ahmed, M. U., Funk, P., Xiong, N. and Folke, M. (2011), 'Case-Based Reasoning Systems in the Health Sciences: A Survey of Recent Trends and Developments', *IEEE Transactions on Systems, Man, and Cybernetics* **41**(4), 421–434.

Dib, E. G., Kidd, M. R. and Saltman, D. C. (2008), 'Case reports and the fight against cancer', *Journal of Medical Case Reports* **2**(1), 39.

Elstein, A. S. and Schwarz, A. (2002), 'Clinical problem solving and diagnostic decision making: selective review of the cognitive literature', *British Medical Journal* **324**(March), 729–732.

Kidd, M. and Hubbard, C. (2007), 'Introducing Journal of Medical Case Reports', *Journal of Medical Case Reports* **1**, 1.

Klein, G. (2008), 'Naturalistic Decision Making', *Human Factors: The Journal of the Human Factors and Ergonomics Society* .

Kolodner, J. L. and Kolodner, R. M. (1987), 'Using Experience in Clinical Problem Solving: Introduction and Framework', *IEEE Transactions on Systems, Man, and Cybernetics* **17**(3).

Manning, C. D., Raghavan, P. and Schütze, H. (2008*a*), Scoring, term weighting and the vector space model, *in* 'Introduction to Information Retrieval', Cambridge University Press, chapter 6, pp. 117–120.

Manning, C. D., Raghavan, P. and Schütze, H. (2008*b*), The term vocabulary and postings lists, *in* 'Introduction to Information Retrieval', Cambridge University Press, chapter 2, pp. 22–27, 32–34.

Mason, R. A. (2001), 'The case report - an endangered species?', *Anaesthesia* **56**(2), 99–102.

Medin, D. L. and Schaffer, M. M. (1978), 'Context Theory of Classification Learning', *Psychological Review* **85**(3), 207–238.

Nissen, T. and Wynn, R. (2014), 'The clinical case report: a review of its merits and limitations', *BioMedCentral Research Notes* **7**(1), 264.

Sandu, N., Chowdhury, T. and Schaller, B. J. (2016), 'How to apply case reports in clinical practice using surrogate models via example of the trigeminocardiac reflex', *Journal of Medical Case Reports* **10**(1), 84.
URL: *http://dx.doi.org/10.1186/s13256-016-0849-z*

Spadaro, S. (2012), ClinicOn Kurzbeschreibung für neue und interessierte Anwender, Technical report.

Williams, D. D. R. (2003), 'In defence of the case report', *The Royal College of Psychiatrists* **184**, 84–88.

Declaration of Authorship

I hereby certify that the work presented here is, to the best of my knowledge and belief, original and the result of my own investigations, except as acknowledged, and has not been submitted, either in part or whole, for a degree at this or any other university.

<div align="right">Osnabrück, May 12, 2017</div>