BEHAVIOURAL SIMULATOR FOR PROFESSIONAL TRAINING
BASED ON NATURAL LANGUAGE INTERACTION

R. Mazza1, L. Ambrosini1, N. Catenazzi1, S. Vanini1, D. Tuggener2, G. Tavarnesi3
1Scuola Universitaria Professionale della Svizzera Italiana (SWITZERLAND)
2University of Zurich, Institut fur Computerlinguistik (SWITZERLAND)
3LifeLike SA (SWITZERLAND)

Abstract

The virtual patient is an online simulation system designed to train and assess relational and clinical abilities in a realistic interactive problem-based learning scenario, where users (medical students) can interact and communicate with characters specifically designed to challenge their clinical and relational skills and facilitate the generation of learning objectives. In this paper we will present an enhancement to the system by simulating a normal interview with real users through natural language, thus enabling users to behave more naturally without keyboards or other input devices. We evaluated the system with a sample of users and found that the new voice-based interaction is user-friendly and facilitates user acceptance. However, a number of limitations remain to be addressed to get the system ready for a large-scale deployment.

Keywords: Natural language interaction, Behavioural Simulator, Medical training.

1 INTRODUCTION

Simulation based learning (SBL) [6] is a rapidly growing paradigm in education. From simple hyperlinked versions of textbooks, digital learning systems have evolved to complete simulation environments where students are faced with complex, real-life like situations. It has been shown that such systems provide definite advantages in terms of learning efficiency [1], [17]. However, current systems are limited by their human-machine interfaces which do not allow for natural interaction between user and simulation.

Traditional interaction with computer systems is usually performed using mouse, keyboard and physical screen. Through reading and interpretation of text and commands the user needs to understand what and how to execute operations, usually by clicking buttons and operating scroll bars. Whereas this way of operating is generally well accepted by back office operators, it causes serious challenges to users not directly involved with computer systems, such as doctors and medical operators.

As explained in section 3, the Virtual Patients are a set of advanced behavioural simulations training systems where final users are medical operators and students who need training in interviewing patients. Currently, interactions between the simulator and the user occur through selection from a set of closed answers using the mouse. Being such interviews conducted between humans in real-life, it is very desirable that the simulation system provides a human-machine interface that mimics natural interaction (i.e. without the limitations of traditional computer system interfaces). In order to enhance the learning efficiency of the system, the objective of our work is to enable interaction with the system using free speech, simulating a normal interview with real patients, thus enabling the user to behave in more natural ways without system constraints linked to the human-machine interface (mouse, buttons etc.). The main objective of this work is the evolution of the Virtual Patient through improvements of the human-machine interface. The introduction of a novel human-machine interface based on natural language aimed to optimise the mode of interaction, enabling users to interact with the system in more natural way, and making simulation more realistic.

2 RELATED WORK

Problem Based Learning (PBL) [21] is an educational strategy adopted in many medical colleges, where students’ learning is guided by the objectives they set themselves in the context of well-planned patient encounter scenarios. A problem is usually presented to students in paper form or in a digital format, and is then investigated and discussed in small groups over two or three sessions.
Currently some PBL tools are available on the market: Campus [7], DecisionSIM\(^1\), Open Labyrinth\(^2\), Web-SP\(^3\). All of them tend to present solutions widely based on static graphic design (such as text-based, cartoons or avatar based learning) and interactive deterministic questionnaires. The typical solution provided is an on-line text-based tool where the user experience is mostly limited to text reading and question-answering. Little pedagogical feedback is given, mainly in the form of correct/wrong answers to the clinical questions. Behavioural analysis and dialogic experience with patient are totally missing. Moreover, none of such systems allow the interaction using natural language. We think that the integration of natural communication into computer-based simulation learning systems will dramatically enhance their learning efficiency.

In particular, extending current simulations – which are currently based on the selection of a number of fixed canned texts – with a component that enable interaction with the system using free speech, allows users to behave in more natural ways, without system constraints linked to the human-computer interface. This component matches a transcribed speech input to a set of available questions. In NLP, this task can be viewed as a similarity ranking problem regarding texts. Modelling semantic similarity of texts is a well-studied and active field in NLP and has gained traction with the advent of Deep Learning-based approaches. One line of work focuses on modelling similarity between words through vector representations of words, so-called word embeddings, which are learnt unsupervised on large text collections [2],[11],[14],[20]. These works are extended into the similarity comparison of sentences, phrases, or whole documents. Texts are embedded as vectors or matrices using variants of recurrent neural networks or convolutional neural networks, often using an attention mechanism [16],[19],[22]. Our work incorporates word embeddings, but also makes use of simpler string matching approaches and term weighting using TF IDF [8],[14].

3 THE VIRTUAL PATIENT

Simulation Based Medical Education (SBME) is an essential part of graduate medical education training. It provides a structured, learner-centred environment in which novice, intermediate, and advanced practitioners can learn or practise skills without causing harm to patients [1]. SBME tools are simulation environments that provide realistic representations of complex clinical environments. They can contribute considerably to improving medical care by boosting medical professionals' performance and enhancing patient safety [23].

The Virtual Patient is a commercial online SBME system, based on artificial intelligence and pre-recorded movies, in which the learner plays the role of a physician who confronts a simulated patient. The patient in the scenario is played by a professional actor who is trained to simulate variable moods, attitudes and emotional responses through verbal and non-verbal communication. The interview can be paused at any point to give time for discussion, generation of learning issues and interaction with the PBL tutor. Virtual Patients are simulators designed to improve users' effectiveness in the areas of anamnesis, diagnosis, treatment and follow-up. Also, the focus is on the process of establishing a trusted relation with the patient. In medical education, they can also be used in a classroom, to trigger the discussion around some learning objectives, under the guidance of a tutor. Each simulation is made by a series of interconnected doctor's visits, where the user can impersonate a General Practitioner or a Specialist, according to the learning objectives of the case. In each visit, the student has to take decisions according to the role and the resource of the doctor he is impersonating. Dialogue interactions between the simulator and the user occur through selection from a set of closed answers, that we call "canned texts", using the mouse (see Figure 1).

At the end of each visit, the system provides a feedback, given directly through the patient’s comments, for example during a post-encounter telephone call between the patient and his daughter or wife, in which he shares his impression about the recent meeting with the doctor. The feedback is based on the decisions and the communication strategy that the user applied during the last visit or visits.

---

1 https://www.kynectiv.com
2 http://openlabyrinth.ca
3 http://websp.lime.ki.se
This work aimed to enhance the learning efficiency of the Virtual Patient, by enabling the interaction with the system using free speech. In order to accomplish this goal, we implemented a NLP component that creates a mapper functionality that selects the most similar canned text given the speech input of the user. If the canned text is identified, then the simulator can use its current algorithm to find an appropriate video response to it. The NLP component is integrated in a new version of the Virtual Patient that includes the mapper functionality and a new user interface (see Figure 2). This prototype was used to run a series of benchmarks and to measure the performance of the NLP components and of the system as a whole, and with a representative number of end-users.

4 NLP INTERACTION

We initially tackled the speech recognition, that is the conversion of spoken language into a text that can be processed by computer. After evaluating several third-party solutions in internal tests, we
settled for Google Cloud Speech API\(^4\). Our evaluation took into account the accuracy of the conversion (as the number of words correctly detected over the number of words contained in the original sentence), the license cost of the solution, and the time required for the speech conversion. For the sake of completeness, we tested speeches recorded by persons with native and non-native English accent (Italians and Germans). One of the results we got from our empirical evaluation was that the REST API of the different solutions have better accuracy than the streaming one. This probably happens because the algorithm for speech conversion can work on the whole sentence and not only on parts of it. For this reason, we chose to record the whole audio data directly from the browser and transmit it to the REST interface of Google Cloud Speech. We also established to use the confidence score returned by the speech recognition software and to discard transcripts having a confidence score below an empirically chosen threshold (as described in section 5).

Although the student now uses a microphone for his interaction with the patient, still, the simulator can only choose among a fixed list of video snippets in order to simulate the patient response, and the speech input cannot directly be mapped to a response video. So, canned texts are still the entry point to the existing system. The solution we adopted was mapping the transcribed speech input to a particular piece of canned text, namely the one that is about the same topic, has the same content, or represents the same dialogue move. If the speech input does not match any canned text, then the system ought to somehow reject the speech input.

The mapping of the transcribed speech input into a particular piece of canned text is performed in two tasks:

1. Rank all available canned texts regarding their similarity to the speech input (question ranking).
2. Check whether the highest ranked canned text is indeed a suitable match for the input (question matching).

For the first task, we employed a machine learning-based approach that takes into account surface features (string overlap ratios) and semantic relatedness based on word embeddings. The first step was to collect training and testing data.

### 4.1 Data collection

To collect the data to train the algorithms, we constructed a version of the system that features a text box input. The student wrote what he wanted to ask the virtual patient into the text box. Next, the available canned texts were shown to the student. He then selected which, if any, of the canned texts was a good match for his input or selected an option that no canned texts matched the input. In experiments with medical students conducted at the Gulf Medical University in Ajman (U.A.E.), we collected around 1000 student inputs of which 60% had a matching canned text in the system.

### 4.2 Question ranking

Using the collected data, we developed and evaluated our approach to question ranking. Due to the nature of the data – pairs of inputs and canned texts and the information whether they match – we formulated the problem as a binary classification task rather than as a ranking problem. We defined the task as follows: Given a transcribed student input and an available canned text: do they match? During the experiment, we performed this binary classification for each student input and the available canned texts. The canned texts were then ranked according to the classification probability of representing a match, and the highest ranked canned text was selected as the match.

We explored three different approaches to tackle the ranking task. First, we applied a simple word overlap metric: Count how many words \( w_i \) are shared between the student input \( q_{nov} \) and a given canned text \( q_{can} \) and normalize by the number of words in the input:

\[
\text{word\_overlap}(q_{nov}, q_{can}) = \frac{|w_i \in q_{nov} \land w_i \in q_{can}|}{|w_i \in q_{nov}|}
\]

The canned text with the most overlapping words was selected as the match. Second, we weighted the words in the student inputs and canned text with TF IDF (based on the canned texts) and used the TF IDF weights to calculate the word overlap. For each word in the student input that is also in the canned text, we added the word’s TF IDF weight to the matching score of the input-canned text pair:

\(^4\) https://cloud.google.com/speech/
Again, the canned text with the highest score was chosen as the match.

Third, we used word embeddings to compare the student input and the canned texts semantically. The motivation was to catch synonym expressions like “what kind of work do you do” - “what is your job”, where we need to recognise that “work” and “job” are synonymous and that the two texts are a good match. To do so, we used pre-trained word embeddings, the Paragram Embeddings [20], to map the words in the inputs and canned texts to their embeddings. We then averaged the embeddings in the input and the embeddings in a given canned text respectively and took the cosine similarity as the score:

\[
\overline{q_{nov}} = \phi(w_t \in q_{nov}), \overline{q_{can}} = \phi(w_t \in q_{can})
\]

\[
\text{similarity}(q_{nov}, q_{can}) = \cosine(\overline{q_{nov}}, \overline{q_{can}})
\]

The canned text with the highest cosine similarity was chosen as the match. Finally, we selected two of the above scores (TF IDF matching score in equation 2 and cosine similarity in equation 4) to train an SVM classifier on the data. We then took the classification probabilities of the SVM to rank the canned texts given a student input.

We evaluated the three ranking approaches on the student inputs that do have a matching canned text. For each student input, we ranked the available canned texts, and if the highest ranked canned texts was indeed the one the student had selected, we counted this as a correct match, and as an incorrect one otherwise. We then calculated accuracy based on the ratio of correct matches given all student inputs. Results are given in table 1. The results show that the simple word overlap metric already gives reasonable results and works even better when TF IDF weights are used. The word embeddings-based approach only marginally surpasses the simple word overlap approach. However, as a feature, it is useful for the SVM classifier which outperforms the other approaches by a large margin.

<table>
<thead>
<tr>
<th>Approach</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random selection</td>
<td>13 %</td>
</tr>
<tr>
<td>Word overlap</td>
<td>71 %</td>
</tr>
<tr>
<td>TF IDF word overlap</td>
<td>76 %</td>
</tr>
<tr>
<td>Word embeddings</td>
<td>72 %</td>
</tr>
<tr>
<td>SVM (5fold crossvalidation)</td>
<td>81 %</td>
</tr>
</tbody>
</table>

4.3 Question matching

In the previous section we introduced and evaluated our approach to question ranking. Our experiments assumed that all student inputs have a corresponding canned text in the system. However, this is an unrealistic assumption, since the student can in theory ask the system anything and the system should react reasonably. In other words, we cannot expect to have a canned text for every student input and thus need to confirm that the highest ranked canned text is indeed a suitable match for the student question.

Our first approach to the task of licensing the highest ranked question as a match for the student input was to use the probability assigned to the question pair by the binary classifier (e.g. reject the match if the probability is below 0.5). However, we found in initial experiments that there is no single suitable threshold to reject a student input-canned text pair.
Therefore, we implemented a filter that licenses question matches consisting of the student input and the highest ranked canned text using high level sentence embeddings combined with a manually crafted feature as inputs for a multi layer perceptron.

The custom feature has been built as a Boolean value which is true if the 3 most important words (calculated using TF-IDF) are included in both the highest ranked question and the student input. Inspired by recent advances in deep learning research we experimented an approach that uses neural networks to build high-level sentence representations, which are then compared using dense layers to obtain a similarity metric. We used word embeddings to transform the text into vectors that can be used to train the neural networks (we used Glove word vectors [14]). We used recurrent neural networks (RNN), more specifically LSTM [5], to capture long dependencies in sentences and to build question embeddings. Those vectors are then concatenated and passed through dense layers to measure their similarity and obtain a metric which is used as a feature. We called this approach Surface-Deep question matching.

Table 2 shows the results obtained with the question matching filter compared to the naive threshold classifier.

<table>
<thead>
<tr>
<th>Approach</th>
<th>Precision</th>
<th>Recall</th>
<th>f1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Threshold</td>
<td>0.85</td>
<td>0.73</td>
<td>0.78</td>
</tr>
<tr>
<td>Surface-Deep question matching</td>
<td>0.84</td>
<td>0.83</td>
<td>0.85</td>
</tr>
</tbody>
</table>

4.4 Sample errors

We conducted some experiments, and we saw that the question matching filter discards almost all student inputs that do not have a matching canned text. However, the NLP matcher still produces some errors. The matcher sometimes selects a wrong canned text that has not relation with a speech input. Also, the filter sometimes rejects matches that are valid. Some examples are reported in Table 3.

<table>
<thead>
<tr>
<th>Approach</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>(q_{nov})</td>
<td>Do you take drugs?</td>
</tr>
<tr>
<td>(q_{nov,sel})</td>
<td>Did you try some substance or drug while you got there?</td>
</tr>
<tr>
<td>(q_{can,best})</td>
<td>Did you take any medications or drugs lately?</td>
</tr>
<tr>
<td>(q_{nov})</td>
<td>Do you have any other problems?</td>
</tr>
<tr>
<td>(q_{nov,sel})</td>
<td>What else? Do you have any chest pain or cough?</td>
</tr>
<tr>
<td>(q_{can,best})</td>
<td>Do you have any other health problems like diabetes or hypertension?</td>
</tr>
</tbody>
</table>

In the first example, the matcher fails to distinguish the two meanings of drug correctly. The student wants to ask whether the patient is a drug user in the sense of substance abuse, while the matcher select the canned text that bears the meaning of drug as a medicine. The ability to computationally determine which sense of a word is triggered by its use in a particular context is called Word Sense Disambiguation (WSD). WSD is known for being a challenging problem [18] and its performance depends on the granularity of the sense distinctions taken into account. The results of comparative evaluations of WSD systems show that when fine-grained sense distinction is employed, accuracy between 65% and 70% can be established in the "all-words task" (disambiguating all class words in a text), whereas better results, between 78 and 81% have been reported in the literature when coarse-grained senses are used [10],[13]. To address this issue, for the first implementation of our VP simulator, we limited to a more coarse-grained view of the senses distinctions and accordingly used canned texts that contain as many words with unique sense as possible.
In the second example, the student input is not specific enough to favour the canned text selected by the student over the one ranked highest by the NLP matcher, i.e. both canned texts are an acceptable match for the student input, but due to the nature of the data only one is selected as the correct one.

Finally, in table 4 we give an example of a valid match that the filter rejected.

<table>
<thead>
<tr>
<th>Approach</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q_{nov}$</td>
<td>&quot;Any other symptoms bothering you? *&quot;</td>
</tr>
<tr>
<td>$q_{can,best}$</td>
<td>&quot;Do you suffer from any other conditions&quot;?</td>
</tr>
</tbody>
</table>

The most important words in the student input are *symptoms* and *bothering* according to the TF IDF ranking. The NLP matcher correctly identified the corresponding canned text. However, when comparing the important words, the cosine similarity between *symptoms* and *condition*, and *bothering* and *suffer* respectively is too low, i.e. the match is rejected.

5 SYSTEM INTEGRATION

Figure 3 depicts the software architecture of the new Virtual Patient simulator. Its main components are:

1. The Client browser, which renders the simulation using HTML5 and JavaScript.
2. The Virtual Patient Server, which hosts the application’s logic and handles the speech recognition and natural language interpretation tasks.
3. The Matcher, which selects the canned text that best matches the input speech.

The information flow between the different components can be described as follow. (1) The audio recording of the student is captured via Javascript and streamed to the Virtual Patient Server (2), which in turn sends it to the Google Cloud Speech (GCS) service via an HTTP post request (3). The transcript returned by GCS is then received by the LifeLike server along with a confidence score (4), which is an estimate between 0.0 and 1.0 of how much the transcript is to be correct. If the confidence is below a threshold (that was empirically determined to be 0.80), the transcript is discarded and the system prompts the student to speak again. Should the confidence score be again below the threshold, the system displays a set of available canned texts and the student selects one of them using the mouse. After this step, if the confidence score is higher than the threshold, the Virtual Patient server builds a JSON object containing the transcript and a list of possible matching canned texts, packed used an XML syntax. The list of candidate canned texts is determined by taking into consideration the step of the interview and by discarding the canned texts previously matched. The JSON object is then forwarded to the Matcher (5) to calculate, for each canned text, the probability to match the transcript. The Matcher returns the scored canned texts by reporting their IDs along with their probability as a JSON object (6), or an error code if there is no-match or an error. These values are then parsed by the Virtual Patient Server to define the ID of the best-matching canned text, which is sent to the Client browser (7). Finally, the Client browser will display the corresponding simulation video (or present an alternative to the user if there is no matching).
The evaluation of the Lifelike virtual patient simulation system was conducted as an iterative process: after an initial informal evaluation carried out by the project team to identify bugs and improvements, the simulator was tested by 8 medical students from two different colleges of medicine: 3 fourth-year students from the University of Insubria, Varese (Italy) and 5 third-year students from the Gulf Medical University, in Ajman (United Arab Emirates, UAE).

The objectives of the test were to collect medical students’ feedback of about new voice-based interaction modality with respect to the previous mouse-based interaction modality. In particular, we intended to investigate whether the new system provided suitable solutions to the typical issues of voice based interface system, that are **discoverability** (how does the user know what to say?) and **learnability** (How easy is it for users to accomplish tasks without previous training?) [3],[9].

We conducted some experiments with the medical students to assess the following:

- check if students easily got familiar with the system and were able to complete a simulation using voice (learnability);
- verify if they were able to “find out” the questions to ask the patients (discoverability);

The tests were also used to get subjective data such as ease of use, satisfaction, user friendliness, recommendations, and suggestions for improvements.

The test was organised in three phases: welcome, user testing, and focus group. The **welcome phase** was dedicated to explaining the reasons of the student involvement, the organisation of the test and to make a demo of the two versions of the simulator. In the **user testing phase** each student was asked to interact with the voice-based version of the virtual patient and complete at least one visit. Each section lasted for about 45 minutes. During this activity, an observer took note of the participant’s behaviour, comments and difficulties. At the end, a **focus group** was carried out involving the students, a moderator and a recorder. Several questions were asked to students concerning their experience using the simulator: opinions, problems, suggestions, and practical applicability in their study programme.

The main results emerged from the user testing and the following focus group are reported below.
In comparing the previous mouse-based interaction with the new voice-based interaction, students preferred the new one, because they found it more realistic and they thought that using pre-defined canned texts limited their opportunity to learn how to correctly interact with a patient.

They were generally able to complete a simulation using voice. They didn’t feel stuck because they usually knew what to say and when. They appreciated that the system showed the textual alternatives only after various failed attempts of communicating by using the voice.

The general reaction of students was very positive:

- they considered the system interesting and enjoyable;
- they appreciated the proposed training approach because it allowed them to interact with a patient earlier than they would have done in their curriculum;
- they liked to be able to learn how to ask questions in the right way and experiment with different communication styles. This was particularly appreciated by some of the participants since they do not receive a specific training on doctor-patient communication;
- the interface was considered clear and clean, there were no doubts about how to start the simulator, conduct the visit, ask for tests, and exit;
- all the users would appreciate using such a tool in their curriculum.

In spite of a general positive opinion, participants found some difficulties during the test and proposed a number of suggestions for improvements.

Some problems were connected to non-matching events: this was because the system did not have answers to some questions that students wanted to ask for a complete anamnesis of the patient (e.g. the question “do you have children?” is not one of the pre-designed ones, so there is not a specific video for the answer) or did not correctly transcribe the speech input (the spoken sentence was not correctly understood by the speech-to-text system because of noise, strong accents or bad pronunciation). These “non-matching” events led the simulator to ask the user to rephrase the last sentence. The ideal solution to solve these issues would be to extend the set of questions the system is able to understand and to use the simulator in a quiet environment, reducing at the most external noise sources.

Participants also encountered some cases of wrong matching: the matcher sometimes selected a wrong canned text that did not have relation with a speech input or didn’t identify a valid canned text, as already revealed by our benchmarks and reported in section 4.4. To minimise this issue, we limited to a more coarse-grained view of the senses distinctions and used canned texts that contained as many words with unique sense as possible. User tests have shown that this restriction is tolerable from a user experience perspective and the system achieves high acceptance.

Another difficulty faced by some students was to find the right question in the right moment; in many cases this was connected to a linguistic issue; the majority of participants thought that the contextual help, which lists by keywords all the topics of the classical medical interview, was enough to suggest the next questions to ask.

Among the proposed improvements, students suggested to have a sort of progress indicator to show where the user is in the simulation process and if he is on the “right track”, and to provide different difficulty levels with or without guidance in conducting the interview.

7 SUMMARY AND CONCLUSION

In this paper we presented an extension of the Virtual Patient, a simulation based medical education tool where medical students can be trained in interviewing patients. The Virtual Patient was extended with a voice-based user interface that allows user to simulate a normal interview with real users through natural language, thus enabling users to behave in more natural ways without keyboards or other input devices.

To allow a natural voice interaction with the Virtual Patient, we built a matching module for the system that maps the speech input into a particular piece of canned text. This was performed in two steps: (1) ranking all available canned texts regarding their similarity to the speech input; (2) checking whether the highest ranked canned text is a suitable match for the input.
We conducted a preliminary evaluation with a small number of medical students to assess the
learnability, discoverability, ease of use, satisfaction, and user friendliness. The participants’ reaction
towards the proposed approach was positive, in spite of some problems still unsolved in the current
version of the voice based Virtual Patient: non-matching situations mainly due to questions asked by
the students that were no part of the set of the canned texts, not-correct matching due to the presence
of similar phases but with a different meaning, and the difficulty, in some cases, to know what question
to ask. Although these problems might represent an obstacle to the wide diffusion of the voice based
Virtual Patient, we believe that the new version of the Virtual Patient is a step ahead toward a more
natural simulation based medical education practice. We are working to solve the problems
encountered in the evaluation and to make the Virtual Patient closer to the needs and to the practice
of medical training.

REFERENCES
mobile voice user interface for accessibility. In Proceeding MobileHCI ’16 Proceedings of the
18th International Conference on Human-Computer Interaction with Mobile Devices and
Services, pages 72-82, 2016.
Potts. A computational approach to politeness with application to social factors. arXiv preprint
Scalese. Features and uses of high fidelity medical simulations that lead to effective learning: a
[7] Cosima Jahnke*, Albrecht Elsasser*, Gudrun Heinrichs, Rudiger Klar, Christoph Bode, and
Thomas K. Nordt. Neue wege in der kardiologischen aus- und weiterbildung. Medizinische
[10] Upali S. Kohomban and Wee Sun Lee. Learning semantic classes for word sense
disambiguation. In Proceedings of the 43rd Annual Meeting on Association for Computational
Linguistics, ACL ’05, pages 34-41, Stroudsburg, PA, USA, 2005. Association for Computational
Linguistics.
February 2009.
word representation. In Empirical Methods in Natural Language Processing (EMNLP), pages
1532/1543, 2014.
[15] Lynne S Robins and Fredric M Wolf. Confrontation and politeness strategies in physician-


