# Personalized symptom checker using medical claims

Sabin Kafle
Cambia Health Solutions, Inc.
Portland, Oregon
sabin.kafle@cambiahealth.com

Penny Pan
Cambia Health Solutions, Inc.
Portland, Oregon
penny.pan@regence.com

Ali Torkamani
Cambia Health Solutions, Inc.
Portland, Oregon
ali.torkamani@cambiahealth.com

Stevi Halley
Cambia Health Solutions, Inc.
Portland, Oregon
stevi.halley@regence.com

John Powers
Cambia Health Solutions, Inc.
Portland, Oregon
john.powers@cambiahealth.com

Hakan Kardes
Cambia Health Solutions, Inc.
Portland, Oregon
hakan.kardes@cambiahealth.com

## ABSTRACT

It is increasingly common for patients to query their symptoms online before approaching medical professionals, with around 1% of Google[1] search queries being related to symptoms [15]. Consequently, building symptom-diagnosis Knowledge Base (KB) and subsequently, symptom checkers is a significant research problem [19], global symptom checkers and online search engines are unable to accommodate personal information which is useful for providing better health recommendations. In this work, we describe our symptom checker which leverages medical claims, demographics, and symptoms to deliver personalized health recommendations. Moreover, we also explain our pipeline for building an integrative KB capable of leveraging both personal and textual information.

## CCS CONCEPTS

• **Applied computing → Health informatics**;

## KEYWORDS

Symptom Checker; Knowledge Base; Personalization

**ACM Reference Format:**
Sabin Kafle, Penny Pan, Ali Torkamani, Stevi Halley, John Powers, and Hakan Kardes. 2018. Personalized symptom checker using medical claims. In *Proceedings of the Third International Workshop on Health Recommender Systems co-located with Twelfth ACM Conference on Recommender Systems (HealthRecSys'18), Vancouver, BC, Canada, October 6, 2018* , 5 pages.

## 1 INTRODUCTION

It is estimated that around 35% of patients' search for their symptoms online before consulting medical personnel according to a survey in 2012 [19]. Symptom checkers and search engines are used primarily to rule out serious conditions and find guidance before seeking physicians. A symptom checker provides diagnostic information based on the symptoms entered by the user. Most symptom checkers also ask the user for personal information including age, gender, and current location to provide more informed medical insights, including nearby medical facilities for treatment of possible ailments. Symptom checkers function by querying users' symptoms to an internal medical KB and then ranking the possible diagnosis using Information Retrieval (IR) methods [11]. The symptoms entered by the user are usually interpreted by a Natural Language Processing (NLP) component to align it to the internal medical KB. User interactions involve either a question answering based approach with questions asked by the symptom checker [8, 12] or a more open textual input including a list down of symptoms and recent events [16].

The vast majority of online symptom checkers are focused on providing a diagnosis based on the symptoms entered by the users. There are some which interact further with a user to obtain additional medical information including any medical history. While the former tends to diagnose without contextual information, the latter suffers from verbosity. Also, users' may not be comfortable in providing their medical history to online services. Another issue also lies in the lack of a robust Natural Language Understanding (NLU) component. While testing out different symptom checkers, most are unable to understand rudimentary paraphrases and negations.

Making a relevant health decision through a symptom checker is based on a reliable internal medical KB [16]. A KB requires human annotation to build accurate relations. Manual annotation is a costly process especially for symptom checkers since it requires efforts from multiple medical professionals to eliminate bias. There have been few efforts to learns KB automatically either using medical texts [10, 16] or Electronic Medical Records (EMRs) [18]. The constructed KBs are heavily refined and validated by medical professionals before usage. Also, no work exists leveraging multiple sources while building a KB, which is essential for more reliable health diagnosis.

In this work, we describe a symptom checker which aims to alleviate some of the shortcomings of currently deployed online symptom checkers. We first describe a medical KB construction pipeline which is capable of leveraging open source medical resources, medical texts[2,3], and medical claims data. Text data are capable of providing medical details which serve as information to an interested user; medical resources enable structure into medical KBs while claims data empowers frequency of diagnosis along with historical information. Secondly, we describe the architecture of the symptom checker with NLP pipeline and personalization as its core component. Our symptom checker has the advantage of

[1] https://www.google.com/

[2] https://www.ncbi.nlm.nih.gov/pubmed
[3] https://en.wikipedia.org/wiki

*HealthRecSys'18, October 6, 2018, Vancouver, BC, Canada*

being able to leverage medical claims into the diagnostic decision resulting in personalized diagnosis (based on historical medical records), recommend providers' specialty and place of service to the user from probable diagnosis.

## 2 RELATED WORK

The earliest version of symptom checkers made predictions for a single or closely related diagnoses (e.g.; breast cancer). Fuzzy rules extracted from neural networks [7] or Bayesian decision rules [9] provide inference from symptoms to diagnosis. More recent symptom checkers mostly describe the KB extraction process with NLU and IR [11] being separate fields.

The KB construction process is a semi-automated method with information extraction tools such as MetaMap [1] used for extraction of medical terminologies. The relations in medical KB are weighted using co-occurrence statistics. This method has found application in Isabel [16] and IBM Watson [10]. Rotmensch et al. [18] describe a method for KB construction based on noisy-OR based Bayesian Networks [13] using Electronic Medical Records (EMRs). Middleton et al. [12] describe a symptom checker which achieves high performance in dataset described by Semigran et al. [19] but requires considerable human effort in building. Reinforcement learning-based question-answer interactions also provide a natural formulation to symptom checkers. Training is performed by conversion of symptom-diagnosis probability mapping to sequences using likelihood sampling [4, 20].

## 3 DATA GENERATION PIPELINE

A significant proportion of work in building a symptom checker lies in the construction of medical KBs. Manual construction of medical KB requires a significant human effort, in turn, making the process expensive. A common alternative is the construction of KB with slight inaccuracies based on medical texts, refined by medical professionals. A generic automated KB construction pipeline requires the following resources[18] - Structured clinical resources (e.g.; UMLS [2], ICD-10 [14]), Medical texts (e.g.; Wikipedia, PubMed[4] abstracts), and Information Extraction (IE) Engine (e.g.; MetaMap [1]). Unified Medical Language System (UMLS) is a medical ontology integrating multiple sources of medical knowledge including SnomedCT [5], ICD-10 using entity defined as concepts to build a hierarchical relation between medical terminologies. SnomedCT is a medical ontology constructed with the objective of defining medical concepts hierarchically. ICD-10 codes are used to describe the diagnosis of patients which is then used by physicians to bill the patient. All the KBs hierarchically describe the concepts with UMLS enabling linkage between multiple KBs.

In addition to the data sources mentioned above, we also use medical claims data. Medical claims give the diagnosis of a patient using ICD-10 codes which can then be cross-referenced with patients personal information to obtain a complete historical picture of a user. The availability of claims data enables construction of a more robust KB which considers temporal dimension as a component of KB. Medical claims also provide a convenient solution for recommending provider specialty and place of service which can be mapped to symptoms using the mapping between symptoms and diagnosis.

We describe our data generation pipeline in the following steps:

(1) Use Wikipedia[5] to obtain textual information regarding ICD-10. Textual information can be attained either through ICD-10 homepage in Wikipedia[6] or using names of ICD-10 code to search in Wikipedia. We use a combination of both to obtain a total of 2,319 diagnosis description linked to ICD-10 diagnosis codes extracted either from the web page or the hierarchical relationship between diagnosis codes.
(2) Use MetaMap to extract all symptoms and diagnosis from PubMed and Wikipedia text. The extracted symptoms and diagnosis are then mapped using co-occurrence statistics between symptoms and diagnosis. To reduce the number of unique diagnosis codes, we use only those diagnosis which has a unique Wikipedia article. All other ICD-10 codes are mapped to the nearest codes using hierarchy relation. Symptoms name are also reduced using name overlap between symptoms to obtain a significantly reduced list. The original list of symptoms can be obtained from UMLS ontology.
(3) Learn the weights in KB between symptoms and diagnosis. We use the Naive-Bayes weight learning [18] to learn associations between symptoms and diagnosis.
(4) Use medical claims data to provide age and gender-based statistics to diagnosis codes, which propagates to symptoms with proportion to learned weights between symptoms and diagnosis. The medical claims are also used to learn the weights between different diagnosis in the temporal dimension. Finally, the medical claims are used to learn provider specialty and place of service for different symptoms based on the learned weights and frequency. We use two year claims data consisting of more than 400k medical claims from around 200k members to build the statistics.

## 4 ARCHITECTURE DESIGN

We summarize our process flow along with architecture in Figure 1. The basic design of symptom checker currently consists of the following components:

- Front-end
- Web server
- Natural Language Processing (NLP) component
- Personalization component

We describe each of the components in detail in preceding subsections.

### 4.1 Front-end

The front-end is the interactive component of the symptom checker where the user interacts with the symptom checker to obtain diagnostic information. It consists of the following two components:

- A query page to obtain the user's symptoms and their personal information (age and gender currently). Users are free to enter additional medical events and any events considered

[4] https://www.ncbi.nlm.nih.gov/pubmed

[5] https://en.wikipedia.org/wiki

[6] https://en.wikipedia.org/wiki/ICD-10

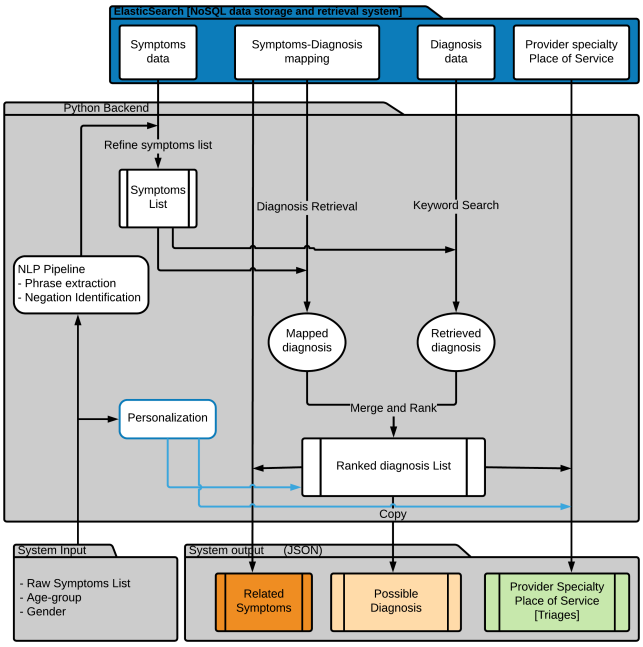


**Figure 1: Process flow for symptom checker with core components**

relevant by the user (e.g., travel to a tropical region before getting symptoms).

- A response page which displays the user's possible conditions, related symptoms, possible place of service and provider specialty, along with a field to include additional symptoms. Figure 2 depicts the response page of the symptom checker.

A design consideration is to make predictions regarding diagnosis regardless of the amount of information entered by the user. The probability score depicts the uncertainty of the model when making predictions.

## 4.2 Web server

The web server is the core component of the system through which the different components of the symptom checker interacts. The symptoms, description, and demographic information entered by the user is processed by the web server. The information is then passed through NLP and personalization component to obtain a better understanding of the symptoms and constraints placed on the possible diagnosis based upon personal information. An ElasticSearch[7] database is then queried to generate candidate diagnosis. The symptom checker then interacts with the database sequentially to further filter and rank the candidates for the entered symptoms and personal information. Then, the personalization component is used to re-rank the diagnosis. The diagnoses are then used to extract useful information including likely place of service, provider

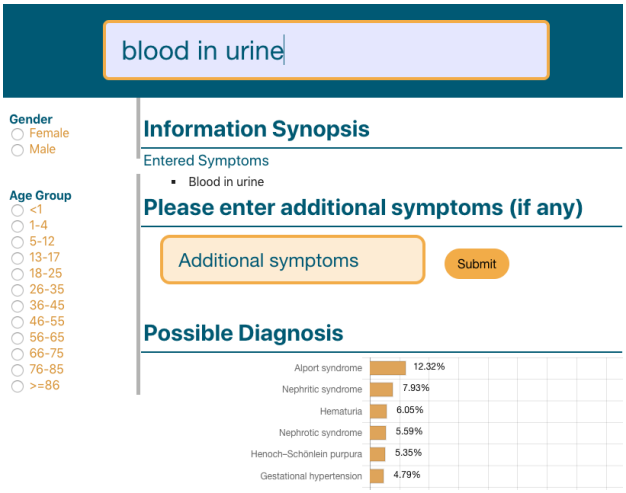


**Figure 2: An example response of the symptom checker**

specialty, and related symptoms specific to the possible conditions. Figure 3 shows the additional information by the symptom checker.

## 4.3 NLP component

NLP processor provides three functionalities - paraphrase generation, negation detection and phrase extractions.

The negation detection and phrase extraction features use the dependency parser based on Spacy Python library[8]. Phrase extraction enables the user to enter symptom checker in either a textual manner with long text input or as a list down of symptoms. Negation detection helps to understand the complex set of information which is useful in ranking out the list of diagnosis based upon both positive symptoms and negative symptoms.

Paraphrase generation component uses Stacked LSTM in an encoder-decoder framework with attention for training similar to [6]. UMLS concepts are used to generate dataset defining medical paraphrases. The dataset provides synonyms for medical phrases including symptoms and diagnosis.

## 4.4 Personalization component

The personalization component enables the symptom checker to provide multiple sets of results for the same symptoms based on personal information of the user. The first step is in identifying the relative importance of symptoms and diagnosis based on the age and gender information of the user. The relative importance is useful for narrowing down the results for symptom checker. The second application of personalization component lies in the re-ranking of the result of symptom checker based on the relevance of the diagnosis to the user based on age, gender, and medical history.

# 5 EVALUATION

We use a dataset of 45 clinical vignettes of different degree of severity of diagnosis described in [19]. A clinical vignette is a full description of a patient condition enabling a physician to make a diagnostic

[7] https://www.elastic.co/products/elasticsearch

[8] https://spacy.io/

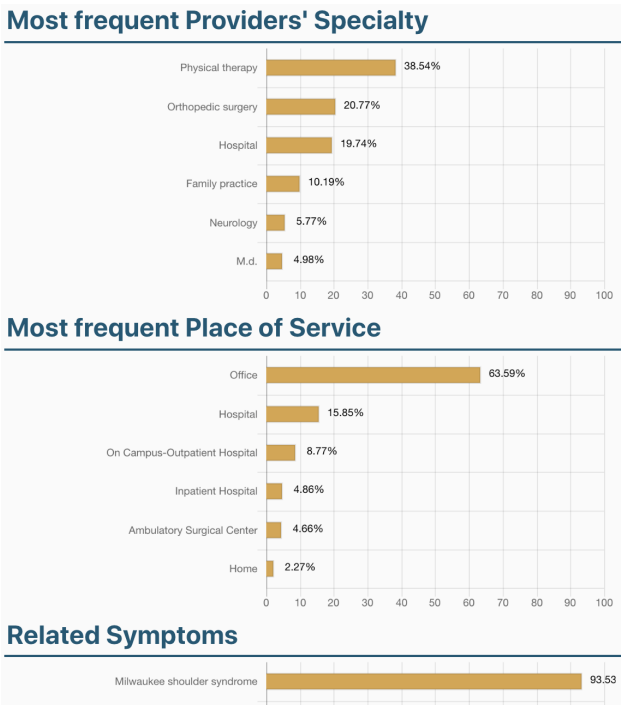


**Figure 3: Additional results generated by the symptom checker for search term "*shoulder pain*".**

decision. The dataset is divided into three degrees of severity - requiring emergent care (15 cases), requiring non-emergent care (15 cases) and requiring self-care only (15 cases). Table 1 lists some example vignettes.

We achieve competitive performance to other online symptom checkers despite using only unsupervised data generation process. We report our accuracy in Table 2. The average performance of symptom checkers is 58% for Top-20 evaluation. The symptoms are manually entered in the format acceptable to the symptom checker to achieve optimal performance. Unlike many online symptom checkers, our system is capable of incorporating noisy text as input and obtaining relevant information through the NLP component.

The accuracy report in Table 2 shows that the symptom checker performs significantly better for diagnosis requiring self-care compared to emergent and non-emergent care. The discrepancy in performance is due to the symptoms listed for self-care conditions having more accurate data source in the form of textual data. A deeper dive is needed to study the discrepancy between medical KB and the evaluation dataset to account for the noise in medical KB and its impact on different care types [3].

Other online symptom checkers as evaluated on Semigran et al. [19] on average obtain 34% Top-1 accuracy and 58% Top-20 accuracy with higher accuracy for emergent care (80%) and least accuracy for self-care (33%) while non-emergent care accuracy is 55%. The performance of some higher quality symptom checkers is higher with Babylon symptom checker [12] obtaining performances similar to medical professionals [17]. The discrepancy in performance is primarily due to the quality of KB with the KB pipeline described

| Diagnosis | Age Gender | Symptoms |
|---|---|---|
| **Requiring Emergent Care** | | |
| **Acute Liver Failure** | 48 y/o Female | • Confusion<br>• Disorientation<br>• Increasingly Drowsy<br>• Mild right upper quadrant pain<br>• Chronic tylenol acetaminophen user - recently took more |
| **Requiring non-emergent Care** | | |
| **Pneumonia** | 6 y/o Male | • History of asthma<br>• Five days fever<br>• Cough<br>• Appetite good<br>• Yellow sputum<br>• Temperature = 101.6 |
| **Requiring self-care** | | |
| **Acute Conjunctivitis** | 14 y/o Male | • 3 days red, irritated eye<br>• Watery discharge from eye<br>• URI symptoms<br>• No pain or light sensitivity |

**Table 1: Examples vignettes extracted from Semigran et al. [19]. There are 15 diagnostic vignettes for each type of care i.e. emergent, non-emergent, and self-care.**

| Metric | Emergent care | Non emergent care | Self-care | Overall |
|---|---|---|---|---|
| Top@1 | 20.00 | 20.00 | 50.00 | 29.55 |
| Top@3 | 20.00 | 33.33 | 57.14 | 36.36 |
| Top@5 | 33.33 | 53.33 | 64.29 | 50.00 |
| Top@10 | 46.67 | 60.00 | 71.43 | 59.09 |
| Top@20 | 53.33 | 66.67 | 78.57 | 65.91 |

**Table 2: Accuracy evaluation (%) of symptom checker across diagnosis requiring emergency, non-emergency and self-care.**

in our system being highly reliant on unsupervised methods rather than being a fully validated medical KB [18]. We expect to obtain better performance on future iterations of our medical KB as we incorporate additional resources and validation methods.

## 6 CONCLUSION AND FUTURE WORK

We have described a symptom checker based upon a medical KB generated in an unsupervised fashion. The novelty of our approach lies in the unsupervised data generation process using multiple data sources, which is then linked with NLP and personalization components to provide a robust, personalized symptom checker. Future work includes refinement of data generation pipeline to integrate additional data sources including EMRs and integration of specific user info into the symptom checker interface to provide a better understanding of individual symptoms.

## REFERENCES


[1] Alan R Aronson. 2006. Metamap: Mapping text to the umls metathesaurus. *Bethesda, MD: NLM, NIH, DHHS* (2006), 1–26.

[2] Olivier Bodenreider. 2004. The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic acids research* 32, suppl_1 (2004), D267–D270.

[3] M Alan Brookhart, Til Stürmer, Robert J Glynn, Jeremy Rassen, and Sebastian Schneeweiss. 2010. Confounding control in healthcare database research: challenges and potential approaches. *Medical care* 48, 6 0 (2010), S114.

[4] Edward Y Chang, Meng-Hsi Wu, Kai-Fu Tang Tang, Hao-Cheng Kao, and Chun-Nan Chou. 2017. Artificial Intelligence in XPRIZE DeepQ Tricorder. In *Proceedings of the 2nd International Workshop on Multimedia for Personal Health and Health Care*. ACM, 11–18.

[5] Kevin Donnelly. 2006. SNOMED-CT: The advanced terminology and coding system for eHealth. *Studies in health technology and informatics* 121 (2006), 279.

[6] Sadid A Hasan, Kathy Lee, Vivek Datla, Ashequl Qadir, Joey Liu, Oladimeji Farri, et al. 2016. Neural Paraphrase Generation with Stacked Residual LSTM Networks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. 2923–2934.

[7] Yoichi Hayashi. 1991. A neural expert system with automated extraction of fuzzy if-then rules and its application to medical diagnosis. In *Advances in neural information processing systems*. 578–584.

[8] Hao-Cheng Kao, Kai-Fu Tang, and Edward Y Chang. 2018. Context-Aware Symptom Checking for Disease Diagnosis Using Hierarchical Reinforcement Learning. (2018).

[9] Igor Kononenko. 2001. Machine learning for medical diagnosis: history, state of the art and perspective. *Artificial Intelligence in medicine* 23, 1 (2001), 89–109.

[10] Adam Lally, Sugato Bagchi, Michael A Barborak, David W Buchanan, Jennifer Chu-Carroll, David A Ferrucci, Michael R Glass, Aditya Kalyanpur, Erik T Mueller, J William Murdock, et al. 2017. WatsonPaths: scenario-based question answering and inference over unstructured information. *AI Magazine* 38, 2 (2017), 59.

[11] Ray R Larson. 2010. Introduction to information retrieval. *Journal of the American Society for Information Science and Technology* 61, 4 (2010), 852–853.

[12] Katherine Middleton, Mobasher Butt, Nils Hammerla, Steven Hamblin, Karan Mehta, and Ali Parsa. 2016. Sorting out symptoms: design and evaluation of the'babylon check'automated triage system. *arXiv preprint arXiv:1606.02041* (2016).

[13] Agnieszka Oniśko, Marek J Druzdzel, and Hanna Wasyluk. 2001. Learning Bayesian network parameters from small data sets: Application of Noisy-OR gates. *International Journal of Approximate Reasoning* 27, 2 (2001), 165–182.

[14] World Health Organization et al. 1992. *The ICD-10 classification of mental and behavioural disorders: clinical descriptions and diagnostic guidelines*. Geneva: World Health Organization.

[15] Veronica Pinchin. 2016. I'm Feeling Yucky :( Searching for symptoms on Google. https://blog.google/products/search/im-feeling-yucky-searching-for-symptoms/

[16] P Ramnarayan, G Kulkarni, A Tomlinson, and J Britto. 2004. ISABEL: a novel Internet-delivered clinical decision support system. *Current perspectives in healthcare computing* (2004), 245–256.

[17] Salman Razzaki, Adam Baker, Yura Perov, Katherine Middleton, Janie Baxter, Daniel Mullarkey, Davinder Sangar, Michael Taliercio, Mobasher Butt, Azeem Majeed, et al. 2018. A comparative study of artificial intelligence and human doctors for the purpose of triage and diagnosis. *arXiv preprint arXiv:1806.10698* (2018).

[18] Maya Rotmensch, Yoni Halpern, Abdulhakim Tlimat, Steven Horng, and David Sontag. 2017. Learning a health knowledge graph from electronic medical records. *Scientific reports* 7, 1 (2017), 5994.

[19] Hannah L Semigran, Jeffrey A Linder, Courtney Gidengil, and Ateev Mehrotra. 2015. Evaluation of symptom checkers for self diagnosis and triage: audit study. *bmj* 351 (2015), h3480.

[20] Kai-Fu Tang, Hao-Cheng Kao, Chun-Nan Chou, and Edward Y Chang. 2016. Inquire and Diagnose: Neural Symptom Checking Ensemble using Deep Reinforcement Learning. In *Proceedings of NIPS Workshop on Deep Reinforcement Learning*.