Decomposing Consumer Health Questions

Kirk Roberts, Halil Kilicoglu, Marcelo Fiszman, and Dina Demner-Fushman
National Library of Medicine
National Institutes of Health
Bethesda, MD 20894
robertske@nih.gov, {kilicogluh, fiszmanm, ddemner}@mail.nih.gov

Abstract

This paper presents a method for decomposing long, complex consumer health questions. Our approach largely decomposes questions using their syntactic structure, recognizing independent questions embedded in clauses, as well as coordinations and exemplifying phrases. Additionally, we identify elements specific to disease-related consumer health questions, such as the focus disease and background information. To achieve this, our approach combines rank-and-filter machine learning methods with rule-based methods. Our results demonstrate significant improvements over the heuristic methods typically employed for question decomposition that rely only on the syntactic parse tree.

1 Introduction

Natural language questions provide an intuitive method for consumers (non-experts) to query for health-related content. The most intuitive way for consumers to formulate written questions is the same way they write to other humans: multi-sentence, complex questions that contain background information and often more than one specific question. Consider the following:

- *Will Fabry disease affect a transplanted kidney? Previous to the transplant the disease was being managed with an enzyme supplement. Will this need to be continued? What cautions or additional treatments are required to manage the disease with a transplanted kidney?*

This complex question contains three question sentences and one background sentence. The focus (*Fabry disease*) is stated in the first question but is necessary for a full understanding of the other questions as well. The background sentence is necessary to understand the second question: the anaphor *this* must be resolved to *an enzyme treatment*, and the predicate *continue*'s implicit argument that must be re-constructed from the discourse (i.e., *continue after a kidney transplant*). The final question sentence uses a coordination to ask two separate questions (*cautions* and *additional treatments*). A decomposition of this complex question would then result in four questions:

1. *Will Fabry disease affect a transplanted kidney?*
2. *Will enzyme treatment for Fabry disease need to be continued after a kidney transplant?*
3. *What cautions are required to manage Fabry disease with a transplanted kidney?*
4. *What additional treatments are required to manage Fabry disease with a transplanted kidney?*

Each question above could be independently answered by a question answering (QA) system. While previous work has discussed methods for resolving co-reference and implicit arguments in consumer health questions (Kilicoglu et al., 2013), it does not address question decomposition.

In this work, we propose methods for automatically recognizing six annotation types useful for decomposing consumer health questions. These annotations distinguish between sentences that contain questions and background information. They also identify when a question sentence can be split in multiple independent questions, and

Proceedings of the 2014 Workshop on Biomedical Natural Language Processing (BioNLP 2014), pages 29–37,
Baltimore, Maryland USA, June 26-27 2014.

when they contain optional or coordinated information embedded within a question.

For each of these decomposition annotations, we propose a combination of machine learning (ML) and rule based methods. The ML methods largely take the form of a 3-step rank-and-filter approach, where candidates are generated, ranked by an ML classifier, then the top-ranked candidate is passed through a separate ML filtering classifier. We evaluate each of these methods on a set of 1,467 consumer health questions related to genetic and rare diseases.

2 Background

QA in the biomedical domain has been well-studied (Demner-Fushman and Lin, 2007; Cairns et al., 2011; Cao et al., 2011) as a means for retrieving medical information. This work has typically focused, however, on questions posed by medical professionals, and the methods proposed for question analysis generally assume a single, concise question. For example, Demner-Fushman and Abhyankar (2012) propose a method for extracting frames from queries for the purpose of cohort retrieval. Their method assumes syntactic dependencies exist between the necessary frame elements, and is thus not well-suited to handle long, multi-sentence questions. Similarly, Andersen et al. (2012) proposes a method for converting a concise question into a structured query. However, many medical questions require background information that is difficult to encode in a single question sentence. Instead, it is often more natural to ask multiple questions over several sentences, providing background information to give context to the questions. Yu and Cao (2008) use a ML method to recognize question types in professional health questions. Their method can identify more than one type per complex question. Without decomposing the full question into its sub-questions, however, the type cannot be associated with its specific span, or with other information specific to the sub-question. This other information can include answer types, question focus, and other answer constraints. By decomposing multi-sentence questions, these question-specific attributes can be extracted, and the discourse structure of the larger question can be better understood.

Question decomposition has been utilized before in open-domain QA approaches, but rarely evaluated on its own. Lacatusu et al. (2006) demonstrates how question decomposition can improve the performance of a multi-sentence summarization system. They perform what we refer to as *syntactic* question decomposition, where the syntactic structure of the question is used to identify sub-questions that can be answered in isolation. A second form of question decomposition is *semantic* decomposition, which can semantically break individual questions apart to answer them in stages. For instance, the question "*When did the third U.S. President die?*" can be semantically decomposed "*Who was the third U.S. President?*" and "*When did X die?*", where the answer to the first question is substituted into the second. Katz and Grau (2005) discusses this kind of decomposition using the syntactic structure, though it is not empirically validated. Hartrumpf (2008) proposes a decomposition method using only the deep semantic structure. Finally, Harabagiu et al. (2006) proposes a different type of question decomposition based on a random walk over similar questions extracted from a corpus. In our work, we focus on syntactic question decomposition. We demonstrate the importance of empirical evaluation of question decomposition, notably the pitfalls of heuristic approaches that rely entirely on the syntactic parse tree. Syntactic parsers trained on Treebank are particularly poor at both analyzing questions (Judge et al., 2006) and coordination boundaries (Hogan, 2007). Robust question decomposition methods, therefore, must be able to overcome many of these difficulties.

3 Consumer Health Question Decomposition

Our goal is to decompose multi-sentence, multi-faceted consumer health questions into concise questions coupled with important contextual information. To this end, we utilize a set of annotations that identify the decomposable elements and important contextual elements. A more detailed description of these annotations is provided in Roberts et al. (2014). The annotations are publicly available at our institution website[1]. Here, we briefly describe each annotation:

(1) BACKGROUND - a sentence indicating useful contextual information, but lacks a question.
(2) QUESTION - a sentence or clause that indicates an independent question.

[1] http://lhncbc.nlm.nih.gov/project/consumer-health-question-answering

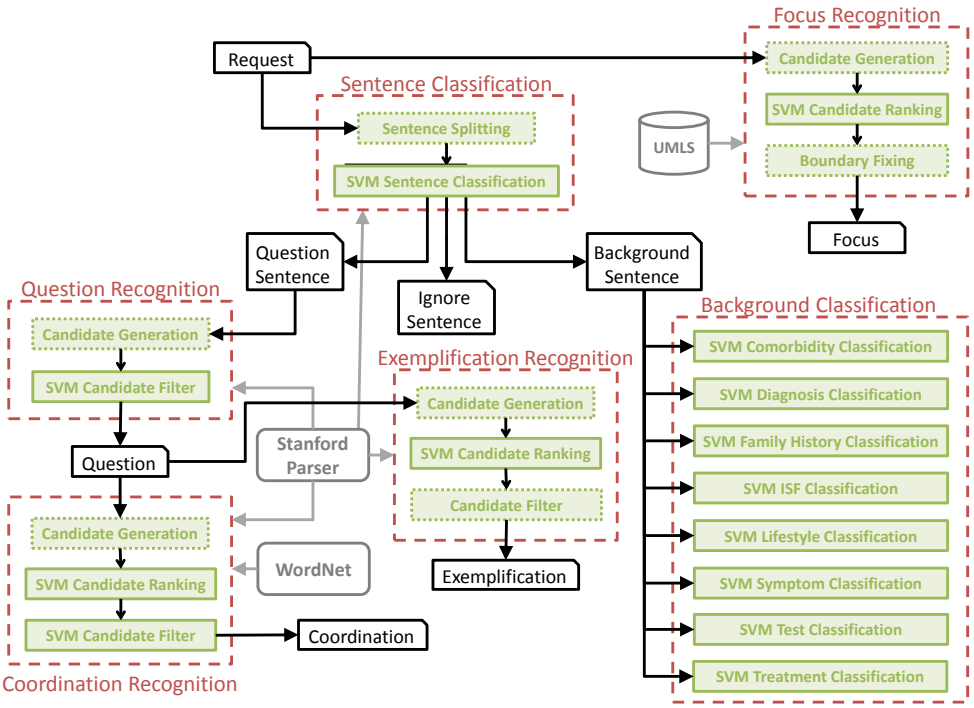

Figure 1: Question Decomposition Architecture. Modules with solid green lines indicate machine learning classifiers. Modules with dotted green lines indicate rule-based classifiers.

(3) COORDINATION - a phrase that spans a set of decomposable items.
(4) EXEMPLIFICATION - a phrase that spans an optional item.
(5) IGNORE - a sentence indicating nothing of value is present.
(6) FOCUS - an NP indicating the theme of the consumer health question.

Further explanations of each annotation are provided in Sections 4-9. To convert these annotations into separate, decomposed questions, a simple set of recursive rules is used. The rules enumerate all ways of including one conjunct from each COORDINATION as well as whether or not to include the phrase within an EXEMPLIFICATION. These rules must be applied recursively to handle overlapping annotations (e.g., a COORDINATION within an EXEMPLIFICATION). Our implementation is straight-forward and not discussed further in this paper. The BACKGROUND and FOCUS annotations do not play a direct role in this process, though they provide important contextual elements and are useful for co-reference, and are thus still considered part of the overall decomposition process.

It should also be noted that some questions are syntactically decomposable, but doing so alters their original meaning. Consider the following two question sentences:

- *Can this disease be cured or can we only treat the symptoms?*
- *Are males or females worse affected?*

While the first example contains two "*Can...*" questions and the second example contains the coordination "*males or females*", both questions are providing a choice between two alternatives and decomposing them would alter the semantic nature of the original question. In these cases, we do not consider the questions to be decomposable.

Data We use a set of consumer health questions collected from the Genetic and Rare Diseases Information Center (GARD), which maintains a website[2] with publicly available consumer-submitted questions and professionally-authored answers about genetic and rare diseases. We collected 1,467 consumer health questions, consisting of 4,115 sentences, 1,713 BACKGROUND sentences, 37 IGNORE sentences, 2,465 QUESTIONS, 367 COORDINATIONS, 53 EXEMPLIFICATIONS, and 1,513 FOCUS annotations. Questions with more than one FOCUS are generally concerned with the relation between diseases. Further information about the corpus and the annotation process can be found in Roberts et al. (2014).

System Architecture The architecture of our question decomposition method is illustrated in

[2]http://rarediseases.info.nih.gov/gard

Figure 1. To avoid confusion, in the rest of this paper we refer to a complex consumer health question simply as a *request*. Requests are sent to the independent FOCUS recognition module (Section 4), and then proceed through a pipeline that includes the classification of sentences (Section 5), the identification of separate QUESTIONS within a question sentence (Section 6), the recognition of COORDINATIONS (Section 7) and EXEMPLIFICATIONS (Section 8), and the sub-classification of BACKGROUND sentences (Section 9).

Experimental Setup The remainder of this paper describes the individual modules in Figure 1. For simplicity, we show results on the GARD data for each task in its corresponding section. In all cases, experiments are conducted using a 5-fold cross-validation on the GARD data. The cross-validation folds are organized at the request level so that no two items from the same request will be split between the training and testing data.

4 Identifying the Focal Disease

The FOCUS is the condition that disease-centered questions are centered around. Many other diseases may be mentioned, but the FOCUS is the disease of central concern. This is similar to the assumption made about a central disease in Medline abstracts (Demner-Fushman and Lin, 2007). Often the FOCUS is stated in the first sentence (typically a BACKGROUND) of the request while the questions are near the end. The questions cannot generally be answered outside the context of the FOCUS, however, so its identification is a critical part of decomposition. As shown in Figure 1, we use a 3-step process: (1) a high-recall method identifies potential FOCUS diseases in the data, (2) a support vector machine (SVM) ranks the FOCUS candidates, and (3) the highest-ranking candidate's boundary is modified with a set of rules to better match our annotation standard.

To identify candidates for the FOCUS, we use a lexicon constructed from UMLS (Lindberg et al., 1993). UMLS includes very generic terms, such as *disease* and *cancer*, that are too general to exactly match a FOCUS in our data. We allow these terms to be candidates so as to not miss any FOCUS that doesn't exactly match an entry in UMLS. When such a general term is selected as the top-ranked FOCUS, the rules described below are capable of expanding the term to the full disease name.

To rank candidates, we utilize an SVM (Fan et al., 2008) with a small number of feature types:

	E/R	P	R	F_1
1st UMLS *Disorder*	E	19.6	19.0	19.3
	R	28.2	27.4	27.8
SVM	E	56.4	54.7	55.6
	R	89.2	86.5	87.9
SVM + Rules	E	74.8	72.5	73.6
	R	89.5	86.8	88.1

Table 1: FOCUS recognition results. E = exact match; R = relaxed match.

- Unigrams. Identifies generic words such as *disease* and *syndrome* that indicate good FOCUS candidates, while also recognizing noisy UMLS terms that are often false positives.
- UMLS semantic group (McCray et al., 2001).
- UMLS semantic type.
- Sentence Offset. The FOCUS is typically in the first sentence, and is far more likely to be at the beginning of the request than the end.
- Lexicon Offset. The FOCUS is typically the first disease mentioned.

During training, the SVM considers any candidate that overlaps the gold FOCUS to be correct. This enables our approach to train on FOCUS examples that do not perfectly align with a UMLS concept. At test time, all candidates are classified, ranked by the classifier's confidence, and the top-ranked candidate is considered the FOCUS.

As mentioned above, there are differences between how a FOCUS is annotated in our data and how it is represented in the UMLS. We therefore use a series of heuristics to alter the boundary to a more usable FOCUS after it is chosen by the SVM. The rules are applied iteratively to widen the FOCUS boundary until it cannot be expanded any further. If a generic disease word is the only token in the FOCUS, we add the token to the left. Conversely, if the token on the right is a generic disease word, it is added as well. If the word to the left is capitalized, it is safe to assume it is part of the disease's name and so it is added as well. Finally, several rules recognize the various ways in which a disease sub-type might be specified (e.g., *Behcet's syndrome vascular type*, *type 2 diabetes*, *Charcot-Marie-Tooth disease type 2C*).

We evaluate FOCUS recognition with both an exact match, where the gold and automatic FOCUS boundaries must line up perfectly, and a relaxed match, which only requires a partial overlap. As a baseline, we compare our results against a fully rule-based system where the first UMLS *Disorder* term in the request is considered the FOCUS.

We also evaluate the effectiveness of our boundary altering rules by measuring performance without these rules. The results are shown in Table 1. The baseline method shows significant problems in precision and recall. It is not able to ignore noisy UMLS terms (e.g., *aim* is both a gene and a treatment). The SVM improves upon the rule-based method by over 50 points in F_1 for relaxed matching. Adding the boundary fixing rules has little effect on relaxed matching, but greatly improves exact matching: precision and recall are improved by 18.4 and 17.8 points, respectively.

5 Classifying Sentences

Before precise question boundaries can be recognized, we first identify sentences that contain QUESTIONs, as distinguished from BACKGROUND and IGNORE sentences. It should be noted that many of the question sentences in our data are not typical wh-word questions. About 20% of the questions in our data end in a period. For instance:

- *Please tell me more about this condition.*
- *I was wondering if you could let me know where I can find more information on this topic.*
- *I would like to get in contact with other families that have this illness.*

We consider a sentence to be a question if it contains any information request, explicit or implicit.

After sentence splitting, we identify sentences using a multi-class SVM with three feature types:

- Unigrams with parts-of-speech (POS). Reduces unigram ambiguities, such as *what-WP* (a pronoun, indicative of a question) versus *what-WDT* (a determiner, not indicative).
- Bigrams.
- Parse tree tags. All Treebank tags from the syntactic parse tree. Captures syntactic question clues such as the phrase tags *SQ* (question sentence) and *WHNP* (wh-word noun phrase).

The SVM classifier performs at 97.8%. For comparison, an SVM with only unigram features performs at 97.2%. While the unigram model does a good job classifying sentences, suggesting this is a very easy task, the improved feature set reduces the number of errors by 20%.

6 Identifying Questions

QUESTION recognition is the task of identifying when a conjunction like *and* joins two independent questions into a single sentence:

- [*What causes the condition*]$_{\text{QUESTION}}$ [*and what treatment is available?*]$_{\text{QUESTION}}$
- [*What is this disease*]$_{\text{QUESTION}}$ [*and what steps can I take to protect my daughter?*]$_{\text{QUESTION}}$

We consider the identification of separate QUESTIONs within a single sentence to be a different task from COORDINATION recognition, which finds phrases whose conjuncts can be treated independently. Linguistically, these tasks are quite similar, but the distinction lies in whether the right-conjunct syntactically depends on anything to its left. For instance:

- *I would like to learn* [*more about this condition* **and** *what the prognosis is for a baby born with it*]$_{\text{COORDINATION}}$.

Here, the right-conjunct starts with a question stem (*what*), but is not a complete, grammatical question on its own. Alternatively, this could be re-formed into two separate QUESTIONs:

- [*I would like to learn more about this condition,*]$_{\text{QUESTION}}$ [**and** *what is the prognosis is for a baby born with it.*]$_{\text{QUESTION}}$

We make this distinction because the QUESTION recognition task requires one fewer step since the boundaries extend to the entire sentence, preventing error propagation from an input module. Further, the features that differentiate our QUESTION and COORDINATION annotations are different.

The two-step process for recognizing QUESTIONs includes: (1) a high-recall candidate generator, and (2) an SVM to eliminate candidates that are not separate QUESTIONs. The candidates for QUESTION recognition are simply all the ways a sentence can be split by the conjunctions *and*, *or*, *as well as*, and the forward slash ("/"). In our data, this candidate generation process has a recall of 98.6, as a few examples were missed where candidates were not separated by one of the above conjunctions.

To filter candidates, we use an SVM with three features types:

- The conjunction separating the QUESTIONs.
- Unigrams in the left-conjunct. Identifies when the left-conjunct is not a QUESTION, or when a question is part of a COORDINATION.
- The right-conjunct's parse tree tag. Recognizes when the right-conjunct is an independent clause that may safely be split.

	P	R	F_1
QUESTION split recognition			
Baseline	24.7	82.4	38.0
SVM	67.7	64.7	66.2
Overall QUESTION recognition			
Baseline	87.3	92.8	90.0
SVM	97.7	97.4	97.5

Table 2: QUESTION recognition results.

For evaluation, we measure both the F_1 score for correct candidates, and the overall F_1 for all QUESTION annotations (i.e., all QUESTION sentences). We also evaluate a baseline method that utilizes the parse tree to recognize separate QUESTIONs by splitting sentences where a conjunction separates independent clauses. The results are shown in Table 2. The baseline method has good recall for recognizing where a sentence should be split into multiple QUESTIONs, but it lacks precision. This is largely because it is unable to differentiate clausal COORDINATIONs such as the above example, as well as when the left-conjunct is not actually a separate question. For instance:

- *Our grandson was diagnosed recently with this disease* **and** *I am wondering if you could send me information on it.*

The SVM-based method can overcome this problem by looking at the words in the left-conjunct. Both methods, however, fail to recognize when two independent question clauses are asking the same question but providing alternative answers:

- *Will this condition be with him throughout his life,* **or** *is it possible that it will clear up?*

While there are methods for handling this issue for COORDINATION recognition, addressed below, recognizing non-splittable QUESTIONs requires far deeper semantic understanding which we leave to future work.

7 Identifying Coordinations

COORDINATION recognition is the task of identifying when a conjunction joins phrases within a QUESTION that can in be separate questions:

- *How can I learn more about [treatments* **and** *clinical trials]*$_{\text{COORDINATION}}$*?*
- *Are [muscle twitching, muscle cramps, and muscle pain]*$_{\text{COORDINATION}}$ *effects of having silicosis?*

Unlike QUESTION recognition, the boundaries of a COORDINATION need to be determined as well as whether the conjuncts can semantically be split into separate questions. We thus use a three-step process for recognizing COORDINATIONs: (1) a high-recall candidate generator, (2) an SVM to rank all the candidates for a given conjunction, and (3) an SVM to filter out top-ranked candidates.

Candidate generation begins with the identification of valid conjunctions within a QUESTION annotation. We use the same four conjunctions as in QUESTION recognition: *and*, *or*, *as well as*, and the forward slash. For each of these, all possible left and right boundaries are generated, so in a QUESTION with 4 tokens on either side of the conjunction, there would be 16 candidates. Additionally, two adjectives separated by a comma and immediately followed by a noun are considered a candidate (e.g., "*a [safe, permanent]*$_{\text{COORDINATION}}$ *treatment*"). In our data, this candidate generation process has a recall of 98.9, as a few instances exist in which a conjunction is not used, such as:

- *I am looking for any information you have about heavy metal toxicity, [treatment, outcomes]*$_{\text{EXEMPLIFICATION+COORDINATION}}$.

To rank candidates, we use an SVM with the following feature types:

- If the left-conjunct is congruent with the highest node in the syntactic parse tree whose right-most leaf is also the right-most token in the left-conjunct. Essentially, this is equivalent to saying whether or not the syntactic parser agrees with the left-conjunct's boundary.
- The equivalent heuristic for the right-conjunct.
- If a noun is in *both*, just the *left* conjunct, just the *right* conjunct, or *neither* conjunct.
- The Levenshtein distance between the POS tag sequences for the left- and right-conjuncts.

The first two features encode the information a rule-based method would use if it relied entirely on the syntactic parse tree. The remaining features help the classifier overcome cases where the parser may be wrong.

At training time, all candidates for a given conjunction are generated and only the candidate that matches the gold COORDINATION is considered a positive example. Additionally, we annotated the boundaries for negative COORDINATIONs (i.e., syntactic coordinations that do not fit our annotation standard). There were 203 such instances in the GARD data. These are considered gold COORDINATIONs for boundary ranking only.

To filter the top-ranked candidates, we use an SVM with several feature types:

	E/R	P	R	F_1
Baseline	E	28.1	36.5	31.8
	R	62.9	75.8	68.7
Rank + Filter	E	38.2	34.8	36.4
	R	78.5	69.0	73.5

Table 3: COORDINATION recognition results. E = exact match; R = relaxed match.

- The conjunction.
- Unigrams in the left-conjunct.
- POS of the first word in both conjuncts. COORDINATIONs often have the same first POS in both conjuncts.
- The word immediately before the candidate. E.g., *between* is a good negative indicator.
- Unigrams in the question but not the candidate.
- If the candidate takes up almost the entire question (all but 3 tokens). Typically, COORDINATIONs are much smaller than the full question.
- If more than one conjunction is in the candidate.
- If a word in the left-conjunct has an antonym in the right conjunct. Antonyms are recognized via WordNet (Fellbaum, 1998).

At training time, the positive examples are drawn from the annotated COORDINATIONs, while the negative examples are drawn from the 203 non-gold annotations mentioned above.

In addition to evaluating this method, we evaluate a baseline method that relies entirely on the syntactic parse to identify COORDINATION boundaries without filtering. The results are shown in Table 3. The rank-and-filter approach shows significant gains over the rule-based method in precision and F_1. As can be seen in the difference between exact and relaxed matching, most of the loss for both the baseline and ML methods come in boundary detection. Most methods overly rely upon the syntactic parser, which performs poorly both on questions and coordinations. The ML method, though, is sometimes able to overcome this problem.

8 Identifying Exemplifications

EXEMPLIFICATION recognition is the task of identifying when a phrase provides an optional, exemplifying example with a more specific type of information than that asked by the rest of the question. For instance, the following contains both an EXEMPLIFICATION and a COORDINATION:

- *Is there anything out there that can help him [such as [medications or alternative therapies]*$_{\text{COORDINATION}}$*]*$_{\text{EXEMPLIFICATION}}$*?*

We could consider this to denote 3 questions:

- *Is there anything out there that can help him?*
- *Is there anything out there that can help him such as medications?*
- *Is there anything out there that can help him such as alternative therapies?*

In the latter two questions, we consider the phrase *such as* to now denote a mandatory constraint on the answer to each question, whereas in the original question it would be considered optional.

EXEMPLIFICATION recognition is similar to COORDINATION recognition, and its three-step process is thus similar as well: (1) a high-recall candidate generator, (2) an SVM to rank all the candidates for a given trigger phrase, and (3) a set of rules to filter out top-ranked candidates.

Candidate generation begins with the identification of valid trigger words and phrases. These include: *especially*, *including*, *particularly*, *specifically*, and *such as*. For each of these, all possible right boundaries are generated, thus EXEMPLIFICATIONs have far fewer candidates than COORDINATIONs. Additionally, all phrases within parentheses are added as EXEMPLIFICATIONs. In our data, this candidate generation process has a recall of 98.1, missing instances without a trigger (see the example also missed by COORDINATION candidate generation in Section 7).

To rank candidates, we use an SVM with the following feature types:

- If the right-conjunct is the highest parse node as defined in the COORDINATION boundary feature.
- If a dependency relation crosses from the right-conjunct to any word outside the candidate.
- POS of the word after the candidate.

As with COORDINATIONs, we annotated boundaries for negative EXEMPLIFICATIONs matching the trigger words and used them as positive examples for boundary ranking.

To filter the top-ranked candidates, we use two simple rules. First, EXEMPLIFICATIONs within parentheses are filtered if they are acronyms or acronym expansions. Second, cases such as the below example are removed by looking at the words before the candidate:

- *I am* **particularly** *interested in learning more about genetic testing for the syndrome.*

In addition to evaluating this method, we evaluate a baseline method that relies entirely on the

	E/R	P	R	F_1
Baseline	E	28.9	62.3	39.5
	R	39.5	84.9	53.9
Rank + Filter	E	60.8	58.5	59.6
	R	80.4	77.4	78.8

Table 4: EXEMPLIFICATION recognition results. E = exact match; R = relaxed match.

syntactic parser to identify EXEMPLIFICATION boundaries and performs no filtering. The results are shown in Table 4. The rank-and-filter approach shows significant gains over the rule-based method in precision and F_1, more than doubling precision for both exact and relaxed matching. There is still a drop in performance when going from relaxed to exact matching, again largely due to the reliance on the syntactic parser.

9 Classifying Background Information

BACKGROUND sentences contain contextual information, such as whether or not a patient has been diagnosed with the focal disease or what symptoms they are experiencing. This information was annotated at the sentence level, partly because of annotation convenience, but also because phrase boundaries are not always clear for medical concepts (Hahn et al., 2012; Forbush et al., 2013).

A difficult factor in this task, and especially on the GARD dataset, is that consumers are not always asking about a disease for themselves. Instead, often they ask on behalf of another individual, often a family member. The BACKGROUND types are thus annotated based on the person of interest, who we refer to as the *patient* (in the linguistic sense). For instance, if a mother has a disease but is asking about her son (e.g., asking about the probability of her son inheriting the disease), that sentence would be a FAMILY_HISTORY, as opposed to a DIAGNOSIS sentence.

The GARD corpus is annotated with eight BACKGROUND types:

- COMORBIDITY
- DIAGNOSIS
- FAMILY_HISTORY
- ISF (information search failure)
- LIFESTYLE
- SYMPTOM
- TEST
- TREATMENT

ISF sentences indicate previous attempts to find the requested information have failed, and are a good signal to the QA system to enable more in-depth search strategies. LIFESTYLE sentences describe the patient's life habits (e.g., smoking, exercise). Currently, the automatic identification of

Type	P	R	F_1	# Anns
COMORBIDITY	0.0	0.0	0.0	23
DIAGNOSIS	80.8	80.3	80.5	690
FAMILY_HISTORY	67.4	38.4	48.9	151
ISF	75.0	65.9	70.1	41
LIFESTYLE	0.0	0.0	0.0	13
SYMPTOM	76.6	48.1	59.1	320
TEST	37.5	4.9	8.7	61
TREATMENT	87.3	35.0	50.0	137
Overall: Micro-F_1: 67.3 Macro-F_1: 39.7				

Table 5: BACKGROUND results.

BACKGROUND types has not been a major focus of our effort as no handling exists for it within our QA system. We report a baseline method and results here to provide some insight into the difficulty of the task.

BACKGROUND types are a multi-labeling problem, so we use eight binary classifiers, one for each type. Each classifier utilizes only unigram and bigram features. The results for the models are shown in Table 5. COMORBIDITY and LIFESTYLE are too rare in the data (23 and 13 instances, respectively) for the classifier to identify. DIAGNOSIS questions are identified fairly well because this is the most common type (690 instances) and because of the constrained vocabulary for expressing a diagnosis. The performance of the rest of the types is largely proportional to the number of instances in the data, though ISF performs quite well given only 41 instances.

10 Conclusion

We have presented a method for decomposing consumer health questions by recognizing six annotation types. Some of these types are general enough to use in open-domain question decomposition (BACKGROUND, IGNORE, QUESTION, COORDINATION, EXEMPLIFICATION), while others are targeted specifically at consumer health questions (FOCUS and the BACKGROUND subtypes). We demonstrate that ML methods can improve upon heuristic methods relying on the syntactic parse tree, though parse errors are often difficult to overcome. Since significant improvements in performance would likely require major advances in open-domain syntactic parsing, we instead envision further integration of the key tasks in consumer health question analysis: (1) integration of co-reference and implicit argument information, (2) improved identification of BACKGROUND types, and (3) identification of discourse relations within questions to further leverage question decomposition.

Acknowledgements

This work was supported by the intramural research program at the U.S. National Library of Medicine, National Institutes of Health. We would additionally like to thank Stephanie M. Morrison and Janine Lewis for their help accessing the GARD data.

References

Ulrich Andersen, Anna Braasch, Lina Henriksen, Csaba Huszka, Anders Johannsen, Lars Kayser, Bente Maegaard, Ole Norgaard, Stefan Schulz, and Jürgen Wedekind. 2012. Creation and use of Language Resources in a Question-Answering eHealth System. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation*, pages 2536–2542.

Brian L. Cairns, Rodney D. Nielsen, James J. Masanz, James H. Martin, Martha S. Palmer, Wayne H. Ward, and Guergana K. Savova. 2011. The MiPACQ Clinical Question Answering System. In *Proceedings of the AMIA Annual Symposium*, pages 171–180.

YongGang Cao, Feifan Liu, Pippa Simpson, Lamont Antieau, Andrew Bennett, James J. Cimino, John Ely, and Hong Yu. 2011. AskHERMES: An online question answering system for complex clinical questions. *Journal of Biomedical Informatics*, 44:277–288.

Dina Demner-Fushman and Swapna Abhyankar. 2012. Syntactic-Semantic Frames for Clinical Cohort Identification Queries. In *Data Integration in the Life Sciences*, volume 7348 of *Lecture Notes in Computer Science*, pages 100–112.

Dina Demner-Fushman and Jimmy Lin. 2007. Answering Clinical Questions with Knowledge-Based and Statistical Techniques. *Computational Linguistics*, 33(1).

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A Library for Large Linear Classification. *Journal of Machine Learning Research*, 9:1871–1874.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.

Tyler B. Forbush, Adi V. Gundlapalli, Miland N. Palmer, Shuying Shen, Brett R. South, Guy Divita, Marjorie Carter, Andrew Redd, Jorie M. Butler, and Matthew Samore. 2013. "Sitting on Pins and Needles": Characterization of Symptom Descriptions in Clinical Notes. In *AMIA Summit on Clinical Research Informatics*, pages 67–71.

Udo Hahn, Elena Beisswanger, Ekaterina Buyko, Erik Faessler, Jenny Traumüller, Susann Schröder, and Kerstin Hornbostel. 2012. Iterative Refinement and Quality Checking of Annotation Guidelines – How to Deal Effectively with Semantically Sloppy Named Entity Types, such as Pathological Phenomena. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation*, pages 3881–3885.

Sanda Harabagiu, Finley Lacatusu, and Andrew Hickl. 2006. Answer Complex Questions with Random Walk Models. In *Proceedings of the 29th Annual ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 220–227.

Sven Hartrumpf. 2008. Semantic Decomposition for Question Answering. In *Proceedings on the 18th European Conference on Artificial Intelligence*, pages 313–317.

Dierdre Hogan. 2007. Coordinate Noun Phrase Disambiguation in a Generative Parsing Model. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 680–687.

John Judge, Aoife Cahill, and Josef van Genabith. 2006. QuestionBank: Creating a Corpus of Parse-Annotated Questions. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 497–504.

Yarden Katz and Bernardo C. Grau. 2005. Representing Qualitative Spatial Information in OWL-DL. *Proceedings of OWL: Experiences and Directions*.

Halil Kilicoglu, Marcelo Fiszman, and Dina Demner-Fushman. 2013. Interpreting Consumer Health Questions: The Role of Anaphora and Ellipsis. In *Proceedings of the 2013 BioNLP Workshop*, pages 54–62.

Finley Lacatusu, Andrew Hickl, and Sanda Harabagiu. 2006. Impact of Question Decomposition on the Quality of Answer Summaries. In *Proceedings of LREC*, pages 1147–1152.

Donald A.B. Lindberg, Betsy L. Humphreys, and Alexa T. McCray. 1993. The Unified Medical Language System. *Methods of Information in Medicine*, 32(4):281–291.

Alexa T McCray, Anita Burgun, and Olivier Bodenreider. 2001. Aggregating UMLS Semantic Types for Reducing Conceptual Complexity. In *Studies in Health Technology and Informatics (MEDINFO)*, volume 84(1), pages 216–220.

Kirk Roberts, Kate Masterton, Marcelo Fiszman, Halil Kilicoglu, and Dina Demner-Fushman. 2014. Annotating Question Decomposition on Complex Medical Questions. In *Proceedings of LREC*.

Hong Yu and YongGang Cao. 2008. Automatically Extracting Information Needs from Ad Hoc Clinical Questions. In *Proceedings of the AMIA Annual Symposium*.