A System for Extracting Study Design Parameters from Nutritional Genomics Abstracts

Cassidy Kelly and Hui Yang

San Francisco State University, 1600 Holloway Ave. San Francisco, CA 94132, USA,
{cassidyk,huiyang}@sfsu.edu

Summary

The extraction of study design parameters from biomedical journal articles is an important problem in natural language processing (NLP). Such parameters define the characteristics of a study, such as the duration, the number of subjects, and their profile. Here we present a system for extracting study design parameters from sentences in article abstracts. This system will be used as a component of a larger system for creating nutrigenomics networks from articles in the nutritional genomics domain. The algorithms presented consist of manually designed rules expressed either as regular expressions or in terms of sentence parse structure. A number of filters and NLP tools are also utilized within a pipelined algorithmic framework. Using this novel approach, our system performs extraction at a finer level of granularity than comparable systems, while generating results that surpass the current state of the art.

1 Introduction

Extracting study design parameters from articles in biomedical journals is an important problem in natural language processing (NLP) that has received a considerable amount of attention in recent years [1, 2, 3, 4, 5, 6, 7, 8, 9]. The task is to locate text fragments and/or numerical values that describe the parameters of the study. These may include the number of subjects, descriptions of the subjects, descriptions of the various treatments, the duration of the study, and any other quantities or descriptions relevant to the design of the study. Algorithms for accomplishing this task play a key role in applications for such tasks as clinical question answering [2], database curation [5], and evidence-based medicine (EBM) [3, 4, 6, 7]. Here we describe algorithms that we have developed for extracting study design parameters from articles in biomedical journals. Specifically, these parameters include the duration of the study; the number of study subjects; and the age, ethnicity, gender, and health status of the study subjects. The algorithms scan sentences to locate the text fragment or numerical value describing the desired parameter. Their implementation will form part of a system for automatically constructing nutrigenomics networks from biomedical journal articles [10, 11]. These networks, drawn from collections of nutritional genomics article abstracts, graphically depict relationships between foods, diseases, chemicals, genes, and proteins, allowing researchers to quickly review summaries of large amounts of published information. Our algorithms will allow users of this system to set values or limits on study design parameters in order to filter the abstracts from which the nutrigenomics networks are created.

In order to extract study design parameters at fine levels of granularity, a number of challenges must be overcome. Some of these are common problems facing NLP applications, such as 1)

the boundary problem [1, 2, 5, 12] and 2) resolving long-distance dependencies [13, 14]. More generally, difficulty in NLP tasks poses a challenge for study design parameter extraction by contributing to 3) imperfect third-party library performance. The nature of the input text poses further challenges, as 4) significant heterogeneity exists in both the writing style of different authors and the nature of different study design parameter types, which are likely to be expressed in dissimilar ways (*e.g.* the age and gender of the study subjects). Even for a single study design parameter type, descriptions will vary significantly from author to author. For instance, some authors express the number of subjects in quantifier phrases (which may be expressed either in letters or numerals); others may simply place parenthetical numbers after the description of each treatment group. Finally, health status may be expressed in myriad ways even within the same article abstract, indeed even within the same sentence. Thus heterogeneity is encountered and must be overcome at several levels. Conversely, 5) homogeneity is also a significant challenge. The various study design parameter types may also be expressed in very similar ways, such as subject age and study duration, which both involve time periods. In general, number disambiguation is a challenge for study design parameter extraction, as numbers are included in descriptors for the subject age, number of subjects, and study duration parameters. Furthermore, in cases where more than one descriptor for the number of subjects is extracted, discrepancies between these numbers must be resolved into a single, final number.

To overcome these challenges, we have designed and implemented a rule-based extraction method, which applies a sequence of various modules in an algorithmic framework to extract study design parameters. The inputs to these rules are sentences known to contain fragments indicating the value of the respective study design parameters; the outputs are the fragments themselves and, where appropriate, a number representing the parameter value. The algorithmic framework is a three-tiered, pipelined architecture for extracting study design parameters. The first pipeline contains preprocessors that annotate the sentence with the output of modules doing named entity recognition and classification (NERC) and parsing. The second pipeline contains modules implementing various extraction rules. The third pipeline contains postprocessors for filtering and refining the extracted parameters.

1.1 Related Work

The problem of recognizing and extracting study design parameter descriptors in unstructured biomedical text falls under the general category of *information extraction*, defined as the recovery of disambiguated, quantifiable information from natural language texts [15]. Within this category, study design parameter extraction has been related to NERC [6]. Techniques for NERC can be differentiated by the degree to which they use labeled training data. Approaches that use no labeled training data rely either on manually crafted rules or unsupervised learning algorithms [12]. If training data are available, supervised or semi-supervised learning algorithms may be used, such as naive Bayes, support vector machines (SVM), hidden Markov models (HMM), maximum entropy models (MEM), and conditional random field (CRF). See Marrero *et al.* [16] or Nadeau *et al.* [12] for more information.

In study design parameter extraction, a number of approaches have been tried. Demner-Fushman and Lin create a comprehensive clinical question-answering system [2], which among other things involves extracting the elements of the PICO framework for formulating clinical queries (*i.e. Patient/Problem*, *Intervention*, *Comparison*, *Outcome*, see [2, 8, 9]). They use a pattern-

matching system to identify candidate fragments describing the study population, and then give the candidates a confidence score based on their position within the abstract and the containing clause. Hara and Matsumoto describe a system for extracting information on the *Patient Population* and *Compared Treatments* from abstracts of confirmatory articles detailing Phase III clinical trials [4]. They employ base noun-phrase (NP) chunking followed by categorization using regular expression matching to identify phrases containing the desired characteristics. Xu *et al.* apply a two-level machine-learning/manual-rules strategy to extract "subject demographics" [7]. Their system first partitions unstructured abstracts into *Background*, *Objective*, *Methods*, *Results*, and *Conclusions* sections with an HMM sequential classifier. It then feeds the *Methods* sentences into an MEM to identify the sentences containing subject demographics. Finally, rules based on hand-crafted grammars and phrase-structure parses are used to identify the exact text strings representing *subject* descriptors, *numbers of trial participants*, and *diseases/symptoms* and their descriptors. Hansen *et al.* use an SVM classifier to classify integers as to whether they represent the number of trial participants in abstracts describing randomized controlled trials [3]. They then assume the largest integer found in this way to be the number of trial participants. Kiritchenko *et al.* present ExaCT, a comprehensive system for the extraction of a large set of study design parameters from full-text articles paired with a user interface for displaying and curating the results [5]. ExaCT first identifies sentences containing the relevant information using an SVM classifier; it then applies "weak" extraction rules to find the text fragments containing the parameters within the identified sentences. Summerscales *et al.* address the problem of automatically extracting summary statistics from abstracts that describe randomized controlled trials [6]. They use a CRF classifier to identify mentions of treatment groups and outcomes, as well as the associated numbers of subjects in each group. Zhao *et al.* develop a system for extracting study design characteristics and elements of the PICO framework to aid the search process for EBM [8, 9]. They use a two-level classification of sentences and fragments where both levels employ machine learning to classify words as belonging to one or more of the following categories: *Sex*, *Age*, *Race*, *Condition*, *Intervention*, and *Study Design*. Their system is closest to ours in terms of the granularity of the study design parameters, but our system extracts contiguous fragments and produces a consolidated number of subjects.

2 Methods

We have created a set of algorithms to extract the following types of study design parameters: *Age of Subjects*, *Duration of Study*, *Ethnicity of Subjects*, *Gender of Subjects*, *Health Status of Subjects*, and *Number of Subjects*. These parameter types were chosen due to their particular relevance in the field of nutritional genomics, but our general technique could be applied to other parameter types as well. Each of our algorithms takes as input a sentence containing the desired parameter type and produces as output the fragments within the sentence that describe that parameter type. For the *Number of Subjects* parameter, a single final number is generated for each sentence as well. The process of creating these algorithms was guided by our intuition and the patterns we observed in a development set of 30 abstracts retrieved from PubMed. Each of these abstracts contains at least one sentence describing one or more of the desired study design parameters.

The extraction algorithms can be divided into those that have to contend with long-distance dependencies and those that do not. This distinction is made clear by the following sentence

```
\b([bB]oys?|[gG]irls?|[mM][ae]n|[wW]om[ae]n|[mM]ales?|[fF]emales?)\b
```

Figure 1: Regular expression for extracting *Gender of Subjects* parameter

from the development set: "The usual intake of soy foods was assessed at baseline, and BP was measured 2-3 y after the baseline survey among 45 694 participants of the Shanghai Women's Health Study aged 40-70 y who had no history of hypertension, diabetes, or cardiovascular disease at recruitment." Here the subjects' health status is described in the relative clause "who had no history of hypertension, diabetes, or cardiovascular disease at recruitment", which is separated from its referent "participants" by nine tokens. The subjects' age is expressed in the participle phrase "aged 40-70 y", itself separated from "participants" by six tokens. The difference is that the word "aged" clearly indicates the expression of an age, while it is the connection to the word "participants" that indicates the relative clause is a health status. In accordance with the division based on long-distance dependencies, the algorithms take one of two basic forms. For those that do not need to identify long-distance dependencies, the extractors consist of regular expression rules. For algorithms that do need to identify long-distance dependencies, the extractors traverse the parse trees created in the preprocessor pipeline.

2.1 Extraction Algorithms Based on Regular Expressions

Many of the extraction algorithms consist primarily of regular expression matching. These extractors utilize information provided by NERC preprocessors, and their outputs are refined by postprocessors. The simplest are based on word lists or gazetteers. For example, the *Gender of Subjects* extraction algorithm matches a short list of gender words, which includes *boys*, *girls*, *men*, *women*, *males*, and *females*. The singular and capitalized forms of these words are also included on the list. The only restriction on this matching is that the matched input text cannot be part of an organization name, which is determined by an NERC module. The regular expression representing this rule is shown in Figure 1. Similarly, the *Ethnicity of Subjects* parameter matches words from a gazetteer. This gazetteer is comprised of demonyms from continents, countries, and one U.S. state (Hawaii), with lists of other adjectives and nouns representing ethnicities also included. Additions include *black*, *Caucasian*, *eastern*, *Hispanic*, *western*, and *white* (as well as inflectional variants). Various restrictions on matches are also enforced. Continent demonyms must not be followed by the words *country* or *nation*. The demonyms *American* and *Canadian* must be preceded by the word *native* or another country demonym. Some of the additions must be followed by a subject word, where *subject* words are from a list originally compiled by Xu *et al.* [7], to which we have made several additions. The matched fragment must not be part of an organization name, as determined by an NERC preprocessor. In addition to the gazetteer-based approach, fragments labeled by the NERC preprocessor as type *nationality* are also extracted. The regular expressions implementing these rules are shown in Figure 2 (without the portion that excludes organization names).

Other study design parameter types utilize more complicated regular expressions based on cue words. The extraction algorithm for the *Age of Subjects* parameter is based on finding number phrases or time period phrases in close proximity to age cue words. A *time period phrase* consists of a number phrase followed immediately by a time unit word. Possible *time unit* words include *hour*, *day*, *week*, *month*, and *year*, plus their plural forms and abbreviations. Numbers are identified by an NERC preprocessor. To facilitate number identification and normalization,

1. `\b((?:[nN]ative|[nN]on)[ -])?(<continent-demonym>)(?! (?:countr(?:y|ies))|nations?)`
2. `\b((?:[nN]ative|[nN]on|(?<!^)[A-Z]+[a-z][a-zA-Z]*)[ -])?(<country-demonym>)([ -]Americans?\b)?`
3. `\b((?:[nN]ative|[nN]on)[ -])?(Hawaiians?)\b`
4. `\b([nN]ative[ -](?:American|Canadian)s?)\b`
5. `\b((?:[bB]lack|[cC]aucasian|[hH]ispanic|[wW]esterner|[wW]hite)s?)\b`
6. `\b((?:[nN]on[ -])?(?:[bB]lack|[cC]aucasian|[eE]astern|[hH]ispanic|[wW]estern|[wW]hite)) (?:populations?|<subject>)\b`
7. `((?<!^)[A-Z]+[a-z][a-zA-Z]*[ -])?(<nationality>)(?! (?:countr(?:y|ies))|nations?)`

Figure 2: Regular expressions for extracting *Ethnicity of Subjects* parameter

dashes separating numbers and words are temporarily changed to spaces (*e.g. 10-year-old* becomes *10 year-old*) and single space separators are changed to commas (*e.g. 10 000* becomes *10,000*). A *number phrase* consists of a single number or two numbers separated only by a dash or the word *to*, while a *time unit* is a token from a closed set of time words and their abbreviations. The algorithm considers a time period phrase in the sentence to be a fragment representing the *Age of Subjects* parameter if there is an age cue word within a four-token neighborhood of the time period phrase. That is, at most three words may occur between the cue word and the time period phrase. The *age* cue words that may precede the time period phrase include *age* and its inflected forms *ages* and *aged*. Those that may follow the time period phrase include *age* and *old*, but only the uninflected forms of each. The regular expressions representing these extraction rules are shown in Figure 3. Matches of these regular expressions may also include various modifiers that influence the interpretation of the numbers.

1. `((?:above|after|average|before|below|between|median|mean|over|under) (?:the )?)?\b(age[ds]?)\W+(?:\S+\s+){0,3}(?:(?:(?:at (?:lea|mo)st |<|>|&(?:(?:#6[02]|[xX]3[cCeE])|[lg]t);)\s*)?)(?<!n ?= ?)(<number>)([ -]to[ -]<number>)?( ?\(<number>\))?([ -]<time>)?`
2. `(at (?:lea|mo)st |<|>|&(?:(?:#6[02]|[xX]3[cCeE])|[lg]t);)?\s*(?<!n ?= ?)(<number>)(?:[ -]to[ -]<number>)?(?: ?\(<number>\))?[ -]<time>\W+(?:\S+\s+){0,3}(age|old)\b`

Figure 3: Regular expressions for extracting *Age of Subjects* parameter

The extraction algorithm for the *Duration of Study* parameter has a complimentary relationship with the extraction algorithm for the *Age of Subjects* parameter. Whereas extraction of the subjects' age involves finding time period phrases near age cue words, extraction of the study duration involves finding time period phrases where age cue words are absent from the nearby context. In other words, the *Duration of Study* extraction algorithm extracts time period phrases that are not within a four-word neighborhood of an age cue word. (Time period phrases modified by the words *each* or *every* are also excluded to avoid extracting dosage intervals.) This rule relies on the *a priori* assumption that the study duration is given somewhere in each input sentence to this algorithm. When such a context occurred in the development set, time period phrases not representing an age were overwhelmingly likely to represent the study duration.

Thus, these two algorithms take advantage of the sentence level context, which has been shown to be sufficiently narrow for reasonable precision to be maintained even for relatively coarse extraction rules [5]. Figure 4 shows the regular expression for *Duration of Study* extraction.

```
(at (?:lea|mo)st |<|>|&(?:(?:#6[02]|[xX]3[cCeE])|[lg]t);)?\s*(?<!\bev
ery )(?<!\b(?:age[ds]?)\W{1,5}(?:\S{1,20}\s{1,5}){0,3})(<number>)(?:
?\(<number>\))?[ -](<time>)(?!  each)(?!\W+(?:\S+\s+){0,3}(?:age|old)
\b)
```

Figure 4: Regular expression for extracting *Duration of Study* parameter

Finally, the extraction algorithm for *Number of Subjects* utilizes two regular expressions. (Other extractors for this parameter type are based on parse information, as detailed below.) One of these matches the common pattern *n* = <*number*>, which is understood to mean the number of study subjects. The other pattern matches numbers followed in close proximity by subject words. These regular expressions are shown in Figure 5.

1. `\b([nN]\s*=\s*<number>)`
2. `(<number>\W+(?:\S+\s+){0,3}(?:<subject>|<demonym>))`

Figure 5: Regular expressions for extracting *Number of Subjects* parameter

2.2 Extraction Algorithms Based on Parse Information

Some types of study design parameters are not as readily identified via pattern-matching rules. Often these parameters are expressed in phrases having long-distance dependency relationships with subject words, whereas regular expressions do better at representing local context. For these parameter types, we have designed extraction rules based on the syntactic structure of the input sentence, which is generated by a parsing module in the preprocessor pipeline. Two types of extractors are based on this parse information. The first utilizes the dependency parse of the sentence. For each subject word in the sentence, as identified by a regular expression, its dependencies are matched against a separate regular expression specific to the parameter type. If no matches are thus found, the dependencies of the dependencies are checked in turn. This process is repeated recursively up to some defined limit in the length of the path, or *dependency chain*. Once a match is found, the text inclusively between the dependent word and the original subject word is extracted.

The *Number of Subjects* parameter is extracted using this technique. It allows a maximum path length of one, searching for dependencies that represent number entities. That is, it searches for direct dependencies of subject words that are numbers. This technique extracts some of the same fragments as the second regular expression in Figure 5, but it also extracts numbers of subjects where the number is more separated from the subject word. This can be the case since numbers precede other modifiers in noun phrases. Consider the following development set sentence: “A total of 400 newly diagnosed breast cancer cases and 400 healthy controls were recruited.” The first number “400” represents a subset of the study subjects, but it is separated from the subject word “cases” by four other words. A dependency parse of the sentence should show that the number “400” is a dependency of the noun “cases”, indicating that it should be extracted for the *Number of Subjects* parameter. Thus, the dependency-chain

method extracts this fragment, while the regular expression would fail. (The second regular expression is included as a fail-safe for when the dependency parse is incorrect.) The *Health Status of Subjects* parameter is also extracted using the dependency-chain method, but with a maximum path length of three. For this parameter type, dependencies must represent *health* entities, which are defined as various semantic types from the UMLS Metathesaurus, generally corresponding to diseases or conditions, found using MetaMap [17]. In this way, we utilize a biomedical vocabulary to create named-entity class at a granularity level appropriate for our algorithm. The specific words *healthy* and *menopause* are also labeled as *health* entities (along with the derivational form *menopausal* and the forms including the prefixes *pre* and *post*) due to their high frequency in *Health Status of Subjects* fragments in the development set.

In cases where the dependency-chain approach fails to extract any parameter fragments, a second technique may be tried. The *minimal-phrase* approach is based on the phrase-structure parse of the sentence. It looks for minimal phrases that contain both a subject word and a match of the parameter-specific regular expression. Starting from the root, each subtree of the phrase-structure parse tree is visited. If a subtree contains both a subject word and a match of the parameter-specific regular expression, and none of its respective subtrees also contain both, the text it spans is extracted. This minimal-phrase approach is used only for the *Health Status of Subjects* parameter. If the set of extracted fragments is empty after applying the dependency-chain extractor, the extraction algorithm *backs off* to the minimal-phrase extractor.

2.3 Additional Processing

After extraction, additional processing may be applied to the extracted fragments in the post-processor pipeline, as various modules are used to refine the results. Extracted fragments that are fully contained within other extracted fragments are removed. Numbers inside fragments are parsed, so that their values can be made programmatically available to other software using our system. Uninformative words are removed from the edges of fragments for the *Health Status of Subjects* parameter. These are words matching a regular expression representing subject words and conjunctions, so that only the health status itself is retained. (Certain subject words are excluded from removal because they do provide additional information; consider the difference between *breast cancer patients* and *breast cancer survivors*.)

The *Number of Subjects* extraction algorithm includes a postprocessor to consolidate the numbers found into a single value representing the total number of subjects. It uses a heuristic based on the intuition that multiple numbers of subjects in a sentence are likely to represent subgroups of study subjects (*e.g.* treatment groups), while the largest number of subjects given in the sentence may represent either a subgroup or the entire population. To determine which of these is the case, the algorithm sums all but the largest number of subjects found. If this sum is equal to the largest number, it is determined to represent the entire population. Otherwise, the largest number is determined to represent another subgroup and is added into the sum to get the final value.

Table 1: Test set partial match statistics

	TP	FP	FN	Prec.	Rec.	F-score
Age of Subjects	16	0	0	1.000	1.000	1.000
Duration of Study	51	4	6	.927	.895	.911
Ethnicity of Subjects	75	2	6	.974	.926	.949
Gender of Subjects	171	0	0	1.000	1.000	1.000
Health Status of Subjects	90	4	22	.957	.804	.874
Number of Subjects	65	2	3	.970	.956	.963

3 Results

To evaluate the extraction algorithms for each of the study design parameter types, a test set of 386 abstracts was used. The abstracts were gathered from PubMed using the query string *soy and cancer*. A human annotator manually tagged the fragments representing the various study design parameters in each abstract using the eHost program[1]. To ensure the correctness of the test data, we subsequently went over the annotations and made necessary corrections, but only after we had finished creating the extraction algorithms (so that the test data did not influence their design). This final annotation was considered to be the gold standard for the test set, resulting in a total of 99 abstracts that included at least one sentence describing the study design parameters, 236 sentences in all.

Using this test set, the extraction algorithm for each type of study design parameter was run on those sentences containing at least one fragment describing the respective parameter type (as annotated by the human reviewer). Thus, the *Age of Subjects* extraction algorithm was run on all and only the sentences in which the subjects' age was described, and similarly for the other parameter types. The output of each algorithm was compared to the human annotator's gold standard. Precision, recall, and F-score were used to quantify the performance. To calculate these measurements, three different criteria were used to define a match between extracted and gold standard fragments (of the same study design parameter type). The *partial match* criterion requires only a partial match between extracted and gold standard fragments, defined as any overlap between the two fragments. The *exact match* criterion considers only an exact match between extracted and gold standard fragments to be a true positive. Any difference in the respective boundaries of the fragments would cause the extracted fragment to be classified as a false positive and the gold standard fragment as a false negative. Lastly, the *word match* criterion determines matches at the word level. Each word contained in both an extracted fragment and a gold standard fragment is considered to be a true positive, whereas any word contained in only an extracted fragment or a gold standard fragment is considered to be a false positive or a false negative, respectively.

The statistics for these three criteria are given in Tables 1, 2, and 3. Primary causes of errors included violations of assumptions built into our rules (*e.g.* that durations preceding *each* represent dosage intervals), simple oversights in the design of the regular expressions (*e.g.* not including capitalized variants), NERC errors (*e.g.* missing diseases expressed as acronyms), and parsing errors (*e.g.* incorrect prepositional phrase attachment).

[1] `http://code.google.com/p/ehost/`

Table 2: Test set exact match statistics

	TP	FP	FN	Prec.	Rec.	F-score
Age of Subjects	14	2	2	.875	.875	.875
Duration of Study	47	8	10	.855	.825	.839
Ethnicity of Subjects	70	7	9	.909	.886	.897
Gender of Subjects	171	0	0	1.000	1.000	1.000
Health Status of Subjects	61	31	50	.663	.550	.601
Number of Subjects	63	4	5	.940	.926	.933

Table 3: Test set word match statistics

	TP	FP	FN	Prec.	Rec.	F-score
Age of Subjects	55	0	8	1.000	.873	.932
Duration of Study	90	10	13	.900	.874	.887
Ethnicity of Subjects	80	2	7	.976	.920	.947
Gender of Subjects	171	0	0	1.000	1.000	1.000
Health Status of Subjects	195	267	86	.422	.694	.525
Number of Subjects	193	7	19	.965	.910	.937

4 Discussion

The previous section gave the evaluation results for each of the study design parameter extraction algorithms. With the exception of the algorithm for *Health Status of Subjects*, these algorithms extracted the desired study design parameter descriptors at very high rates of success. They also achieved good balance between precision and recall, demonstrating that overfitting of the development set was minimized. As for the *Health Status of Subjects* parameter, its extraction algorithm still produced fragments which nearly always contained health status information, and a substantial percentage of the time produced the exact desired fragments. We would argue that in cases where it returned a large number of incorrect words (as evidenced by the low precision numbers in Table 3), its output still represents an improvement over algorithms which simply return the entire sentence.

We now compare our results with comparable results reported in the related work. However, it must be noted that none of these comparisons represent a perfect match in the experimental setup which generated the results. Xu *et al.* report an accuracy of 92.5% for extracting the total number of trial participants from sentences [7], which is most comparable to the output of our *Number of Subjects* consolidation heuristic. The accuracy of this algorithm was 82.7%. Since this value was calculated using only those sentences in which the number of subjects was given (implicitly or explicitly), the number of true negatives used to calculate it was zero. It is unclear if the accuracy calculation done by Xu *et al.* for the number of trial participants included any true negatives. They also report an accuracy of 82.5% for extracting *subject* descriptors, which would likely include many of the study design parameters extracted by our algorithms. However, with the extraction granularity so much coarser, this number is not comparable to those produced by our algorithms. Hansen *et al.* report an accuracy of 97.03% for finding the correct number of trial participants in 75 abstracts [3]. Since this accuracy is given per abstract, it is not as applicable of a comparison as the number given by Xu *et al.*. However, they also report the precision, recall, and F-score for classifying individual integers; these values are .97, .74, and .84, as compared with .940, .926, and .933 for our *Number of Subjects* extraction algorithm under the exact match criterion.

Kiritchenko *et al.* report several comparable statistics from the ExaCT program [5], specifically for the *sample size* and *duration of treatment* information elements. The *eligibility criteria* information element is similar to the other study design parameters that our extraction algorithms target, but at a coarser level of granularity and with no fragment-level extraction done. For *sample size*, they report exact match precision and recall of .89 and .87, respectively, as compared with exact match precision and recall of .940 and .926 for our *Number of Subjects* extraction algorithm. Their partial match results are the same for *sample size*, but our results for *Number of Subjects* increase to .970 and .956 under the partial match criterion. For *duration of treatment*, Kiritchenko *et al.* report exact match precision and recall of .84 and .91, respectively, whereas our exact match precision and recall for *Duration of Study* are .855 and .825. For partial matches, their *duration of treatment* precision and recall rise to .86 and .93, while ours for *Duration of Study* increase to .927 and .895.

Summerscales *et al.* achieve a precision/recall/F-score of .82/.77/.80 for extracting numbers representing group sizes [6], whereas our extraction algorithm for *Number of Subjects* achieve precision/recall/F-score of .940/.926/.933. Finally, Zhao et al. provide word-level classification results for the categories *Sex*, *Age*, *Race*, and *Condition* [8, 9]. We compare these results to our own word-level results. For *Sex*, their best precision/recall/F-score (from two articles) are .98/1.00/.99 as compared with 1.000/1.000/1.000 for our *Gender of Subjects* algorithm. For *Age*, their best numbers are .85/.78/.81 as compared with 1.000/.873/.932 for our *Age of Subjects* algorithm. For *Race*, their best are .92/.86/.89 as compared with .976/.920/.947 for our *Ethnicity of Subjects* algorithm. For *Condition*, their best are .76/.63/.69, as compared with .422/.694/.525 for our *Health Status of Subjects* algorithm. We note that the mapping between *Condition* and *Health Status of Subjects* is significantly weaker than the other cases.

4.1 Conclusion

While the differing natures of the algorithms and test conditions make direct comparison difficult, our system produces results that meet or exceed the most comparable reported results described above, while at the same time performing extraction at the finest level of granularity yet seen. In doing so, we have demonstrated that manually designed rules can produce results comparable or superior to those currently achievable through the use of machine learning. Most of our extraction algorithms can be expressed in terms of standard NLP tasks performed by preprocessor modules, regular expressions, and simple filtering algorithms performed by postprocessor modules, all tied together in the algorithmic framework that we have created. For those study design parameter types whose extraction algorithms additionally require parse information, we have designed the algorithms in such a way as to be easily expressible in terms of phrase-structure and dependency grammar. In addition to these primary contributions, we have also created a useful heuristic for determining the consolidated number of subjects indicated by multiple extracted fragments. Our system is modular, efficient, and high-performance. Because of its simplicity, it can be readily expanded or improved (*e.g.* by updating or adding new regular expressions). However, even without any adjustments, our system's performance will improve simply by utilizing the new state-of-the-art tools for parsing and NERC as they are developed in the future. Thus, we assert that our system is the new state of the art for study design parameter extraction.

References

[1] B. de Bruijn, S. Carini, S. Kiritchenko, J. Martin and I. Sim. Automated Information Extraction of Key Trial Design Elements from Clinical Trial Publications. *AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium*, pages 141–145, 2008.

[2] D. Demner-Fushman and J. Lin. Answering Clinical Questions with Knowledge-Based and Statistical Techniques. *Comput. Linguist.*, 33(1):63–103, 2007.

[3] M. Hansen, N. Rasmussen and G. Chung. Extracting Number of Trial Participants from Abstracts of Randomized Controlled Trials. In *Troms Telemedicine and eHealth Conference*. 2008.

[4] K. Hara and Y. Matsumoto. Extracting Clinical Trial Design Information from MEDLINE Abstracts. *New Generation Computing*, pages 263–275, 2007.

[5] S. Kiritchenko, B. de Bruijn, S. Carini, J. Martin and I. Sim. ExaCT: automatic extraction of clinical trial characteristics from journal publications. *BMC Medical Informatics and Decision Making*, 10(1):56, 2010.

[6] R. L. Summerscales, S. Argamon, S. Bai, J. Hupert and A. Schwartz. Automatic summarization of results from clinical trials. In *Proceedings of the 2011 IEEE International Conference on Bioinformatics and Biomedicine*, BIBM '11, pages 372–377. IEEE Computer Society, Washington, DC, USA, 2011.

[7] R. Xu, Y. Garten, K. S. Supekar, A. K. Das, R. B. Altman and A. M. Garber. Extracting Subject Demographic Information From Abstracts of Randomized Clinical Trial Reports. In *12th World Congress on Health (Medical) Informatics*. 2007.

[8] J. Zhao, M. yen Kan, P. M. Procter, S. Zubaidah, W. K. Yip and G. M. Li. Improving Search for Evidence-based Practice using Information Extraction. In *AMIA 2010 Annual Symposium*. 2010.

[9] J. Zhao, P. Bysani and M. yen Kan. Exploiting Classification Correlations for the Extraction of Evidence-based Practice Information. In *AMIA 2012 Annual Symposium*. 2012.

[10] H. Yang, A. Sharma, R. Swaminathan and V. Ketkar. On Building a Quantitative Food-Disease-Gene Network. In *2nd International Conference on Bioinformatics and Computational Biology (BICoB)*. 2010.

[11] H. Yang, R. Swaminathan, A. Sharma, V. Ketkar and J. D'Silva. Mining Biomedical Text towards Building a Quantitative Food-Disease-Gene Network. In M. Biba and F. Xhafa (editors), *Learning Structure and Schemas from Documents*, volume 375 of *Studies in Computational Intelligence*, pages 205–225. Springer Berlin Heidelberg, 2011.

[12] D. Nadeau and S. Sekine. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1):3–26, 2007.

[13] J. R. Finkel, T. Grenager and C. Manning. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 363–370. Association for Computational Linguistics, Stroudsburg, PA, USA, 2005.

[14] J. Gao and H. Suzuki. Capturing Long Distance Dependency in Language Modeling: An Empirical Study. In *Proceedings of the First international joint conference on Natural Language Processing*, IJCNLP'04, pages 396–405. Springer-Verlag, Berlin, Heidelberg, 2005. URL http://dx.doi.org/10.1007/978-3-540-30211-7_42.

[15] H. Cunningham. Information Extraction, Automatic. In *Encyclopedia of Language and Linguistics*. Elsevier, 2 edition, 2005.

[16] M. Marrero, S. Snchez-Cuadrado, J. Morato Lara and G. Andreadakis. Evaluation of Named Entity Extraction Systems. *Research In Computer Science*, 41:47–58, 2009.

[17] A. R. Aronson. Effective Mapping of Biomedical Text to the UMLS Metathesaurus: The MetaMap Program. In *Proceeding of the 2001 Annual Symposium of the American Medical Informatics Association (AMIA 2001)*, pages 17–21. 2001.