Test suite design for ontology concept recognition systems

K. Bretonnel Cohen[1,2], Christophe Roeder[1], William A. Baumgartner Jr.[1], Lawrence E. Hunter[1], Karin Verspoor[1]

1: Center for Computational Pharmacology
University of Colorado School of Medicine
Aurora, Colorado, USA
2: Department of Linguistics
University of Colorado at Boulder
Boulder, Colorado, USA
Corresponding author: kevin.cohen@gmail.com

Abstract

Systems that locate mentions of concepts from ontologies in free text are known as ontology concept recognition systems. This paper describes an approach to the evaluation of the workings of ontology concept recognition systems through use of a structured test suite and presents a publicly available test suite for this purpose. It is built using the principles of descriptive linguistic field work and of software testing. More broadly, we also seek to investigate what general principles might inform the construction of such test suites. The test suite was found to be effective in identifying performance errors in an ontology concept recognition system. The system could not recognize 2.1% of all canonical forms and no non-canonical forms at all. Regarding the question of general principles of test suite construction, we compared this test suite to a named entity recognition test suite constructor. We found that they had twenty features in total and that seven were shared between the two models, suggesting that there is a core of feature types that may be applicable to test suite construction for any similar type of application.

1. Introduction

Ontologies of the biomedical domain have become a crucial enabling technology for modern molecular biology. Currently at least 160 biomedical ontologies exist or are under construction. One major focus of current genomic science is the construction of enormous databases of genes and proteins in which the objects of description are labelled with concepts from one or more ontologies that describe such things as their molecular function, subcellular location, and biological processes that they participate in. Creating tools to assist with these databasing activities has been a major focus of much recent work in BioNLP, or biomedical text mining (Hirschman et al., 2005; Krallinger et al., 2008).

One way in which it has been proposed that text mining can assist the database curators is by building tools that automatically mine associations between genes or proteins and concepts from the many ontologies of the biomedical domain (Blaschke et al., 2005). To do this, it is necessary to be able to accomplish two tasks: locating mentions of genes or proteins in text and associating them with unique database identifiers, known as gene normalization (Hirschman et al., 2005; Morgan et al., 2007; Morgan et al., 2008); and locating concepts from ontologies and associating those mentions with genes. Systems that locate mentions of concepts from ontologies in free text are known as ontology concept recognition systems.

In general, performance on the two tasks in concert has been low. The aggregate task was part of the BioCreative I shared task, and the best system performance was a precision of 34.62. (Recall was not measured but was clearly very low—the highest-precision system extracted only 26 relations.) One cause of this low performance was undoubtedly the difficulty of associating mentions of genes with unique database identifiers, and another was associating mentions of genes with mentions of ontology concepts. However, recognizing ontology concepts was itself a clear cause of errors in the systems that participated. (Camon et al., 2005) gives anecdotal examples of some system errors.
This paper describes an approach to the evaluation of the workings of ontology concept recognition systems through use of a structured test suite and presents a publicly available test suite for this purpose. A structured test suite is built by enumerating the factors that might affect system performance and then systematically isolating, varying, and combining them in a data set in which inputs are paired with gold standard outputs. The goal is to build a test suite to which arbitrary ontology concept recognition systems can be applied.
More broadly, we also seek to investigate what general principles, if any, might inform the construction of such test suites. We return to this topic in the Discussion section.

1.1. Related work

(Cohen et al., 2004) demonstrated that principles of descriptive linguistics (Bloomfield, 1933; Harris, 1951; Gleason Jr., 1961; Samarin, 1967) and software testing (Beizer, 1990; Beizer, 1995; Binder, 1999; Kaner et al., 1999; Patton, 2005; Kaner et al., 2002; Myers, 1979; Marick, 1997) could be applied to build a system for the dynamic generation of structured test suites for biomedical gene/protein named entity recognition (hereafter GM, or gene mention) systems. They applied a simple test suite to five GM systems and found that the test suite revealed errors in all five systems. They also attempted to predict performance on standard NER metrics for specific equivalence classes for one GM system, and found that four of five predictions were verifiable. However, there was no attempt to gener-

alize the work to tasks other than gene/protein named entity recognition. In contrast to that effort, the work presented here attempts to build a test suite for a much broader and less clearly defined class of semantic entities—namely, concepts from a broad ontology.

More generally, outside of the biomedical domain, the subject of test stuites has been studied in the grammar engineering community, resulting in such efforts as the early Hewlett-Packard syntax test suite and the more recent TSNLP project (Oepen et al., 1998; Volk, 1998). Those efforts have been restricted to testing the handling of morphosyntactic features; the work presented here tests the hypothesis that test suites are applicable to a considerably wider range of applications than syntactic parsers.

Other work on evaluating ontology concept recognition systems has focussed less on granular evaluation of performance than on system comparison. (Shah et al., 2009) compared the MetaMap and Mgrep systems for locating mentions of concepts in four genres of text, finding that Mgrep generally outperformed MetaMap in terms of precision. Their evaluation was done by post-hoc manual examination of outputs by four domain experts, making it difficult to repeat the experiment. In contrast, our test suite can be run many times a day, making it useful for such tasks as systematic exploration of the parameter space of multiple concept recognizers. Our design also made it easy to evaluate performance on all three sub-ontologies of the Gene Ontology, while (Shah et al., 2009) only examined performance on the biological process sub-hierarchy (along with three other independent ontologies). We expand on the advantages of our approach versus post-hoc analysis such as theirs in the Discussion section.

In the area of locating ontology concepts in text, the work of (Verspoor et al., 2003) is relevant. They found that only 6% of Gene Ontology terms occurred verbatim in the corpus; this highlights the importance of the test cases in which we change the form of terms.

2. Materials and Methods

2.1. Materials

As the target for our test suite construction efforts, we used the version of the Gene Ontology available on 9/24/2009 at 4:28 PM Mountain time.

Because we do not believe in punishing system developers for graciously making their applications freely available, we do not identify here the concept recognition system that we used in our experiments. It is freely publicly available and is typical of other ontology concept recognition systems.

2.2. Methods

As was demonstrated in (Cohen et al., 2004), test suites for GM can be built based on basic analytical techniques of descriptive linguistic field work and on the principles of software testing, which are shown there to have much in common. We applied a similar approach to the design of our test suite here. The specific factors that we considered were mostly linguistically motivated. Others were motivated by the structure of modern biomedical ontologies, and others were motivated by our knowledge of common techniques in the construction of ontology concept recognition systems. The dimensions that we considered are classifiable into two broad categories: features of the terms themselves, and types of changes in the terms. The dimensions and their categories are as follows:

2.2.1. Features of the terms

- Length
- Punctuation
- Presence of stopwords
- Ungrammatical terms
- Presence of numerals, Arabic and Roman
- Official synonyms
- Ambiguous terms

2.2.2. Types of changes in the terms

- Singular/plural variants
- Ordering and other syntactic variants
- Inserted text
- Coordination
- Verbal versus nominal constructions
- Adjectival versus nominal constructions
- Unofficial synonyms

In order to understand the architecture of the test suite, it is helpful to understand the structure of concepts in modern biomedical ontologies. (We recognize that the term "concept" is controversial here, but use it to accord with much other work in biomedical ontologies.) Each concept is a triple of a unique identifier, a term, and a definition. Optionally, they may also have synonyms associated with them. All terms are nominals—there are no verbs (at least in the Gene Ontology). A typical concept from the well-known Gene Ontology is the following:

```
Identifier: GO:0005623
Term: cell
Definition: The basic structural
and functional unit of all organisms.
Includes the plasma membrane and
any external encapsulating
structures such as the cell wall and
cell envelope.
```

The test suite was constructed by locating concepts in the ontology whose terms had particular characteristics—of length, singularity/plurality, etc. Inputs consist of either terms or of the results of specific changes applied to those terms, as described below. Each term was then paired with the identifier that a system should return when that term is encountered.

As mentioned above, some inputs consisted of terms exactly as they appeared in the ontology. This might seem a

trivial test, but it is a necessary sanity check on the performance of the concept recognition system, and as we will see, the system under test failed to recognize even the canonical forms of some terms. Other inputs were constructed by applying some change, motivated either by linguistic knowledge or by insight into typical system construction, to a term. An example of this is pluralization of singular terms. For example, our test suite contains the test cases

```
ID: GO:0005623 cell
ID: GO:0005623 cells
```

. . . separated into two equivalence classes, one for singulars and one for plurals. Note that in each case, the identifier that should be returned is the same, but only the singular form is the canonical term for the concept.

2.3. Specifics of the equivalence classes and transformations

2.3.1. Features of terms

Length: Concepts were selected with terms whose lengths varied from one to ten tokens. To the greatest extent possible, we avoided having any of those tokens be either stop words, numerals, or single letters, since those form equivalence classes of their own. We picked this axis to vary both for linguistic reasons—length is known to be relevant for the operation of many linguistic rules, ranging from tone assignment in tone languages to the formation of comparatives and of irregular past tenses in English—and due to insight into system construction. Length of gene names has previously been shown to be important to the performance of GM systems (Kinoshita et al., 2005; Yeh et al., 2005) and gene normalization systems (Hirschman et al., 2005). It is also of clear relevance to ontology concept recognition, since we know that some systems operate via sliding windows of limited size.

Punctuation: Concepts were selected whose terms contained punctuation, including *(),-'*. These were included because of the likelihood that shallow parsers would be thrown off by punctuation that normally demarcates syntactic units.

Presence of stopwords: Concepts were selected whose words contained tokens that are commonly on stopword lists. These are known to create problems for information-retrieval-based approaches to concept recognition, such as (Johnson et al., 2006; Johnson et al., 2007). We also hypothesized that they might pose problems for systems based on shallow parsing of noun phrases.

Ungrammatical terms: Ontologies have been pointed out to contain terms that are not themselves grammatical in English, such as *phagocytosis, recognition* (Ceusters et al., 2005), and these have clear implications for text-based methods of locating ontology concepts in text.

Presence of numerals, Arabic or Roman: This category was included because it is known to be an unusual feature of noun phrases (see e.g. the low or nonexistent coverage of this phenomenon in (Quirk et al., 1985; Biber et al., 1999)). We also found in our work on GM that such inputs were handled erroneously in one system.

Official synonyms: Some terms in some ontologies have synonyms associated with them. These should return the same concept identifier as the official term.

Ambiguous terms: In previous work, we have found that some ontologies contain ambiguous terms (Johnson et al., 2007). For example, many terms in the ChEBI ontology (Degtyarenko, 2003) contain synonyms which are ambiguous—*C* is a synonym of four different concepts in ChEBI.

2.3.2. Features of variation in the terms

Singular/plural: Concepts were selected whose terms were singular and which had both regular and irregular plurals, separated into different equivalence classes. We also selected concepts whose terms contained a plural, such as GO:0007272 *ensheathment of neurons*.

Ordering and other syntactic variants: Many terms are susceptible to variability in ordering and in other aspects of syntactic realization, such as paraphrases that have function words inserted or deleted, e.g. *apoptosis induction* instead of the canonical GO:0006917 *induction of apoptosis*.

Inserted text: This feature deals with the common phenomenon of the presence of other material within the boundaries of a term in free text, e.g. *some* in *ensheathment of some neurons*, derived from the canonical form GO:0007272 *ensheathment of neurons*.

Coordination: This is a special case of syntactic variability, included because of its prevalence in free text and its difficulty for any language processing system.

Verbal versus nominal constructions: As pointed out above, all terms in the prototypical ontology are nominals. However, ontology concepts can appear in verbal forms. For example, GO:0016477 *cell migration* might appear in free text as *cell migrated*. We included all inflectional forms of the corresponding verbs in the test suite.

Adjectival versus nominal forms: Similarly, many nominals, including some that are high-frequency in this domain, can occur in adjectival forms. A common example is GO:0005634 *nucleus*, which often appears as *nuclear*.

2.4. Applying the concept recognizer

Once the test suite was constructed, we applied the concept recognition system to it. We mostly used the default parameter settings.

3. Results

3.1. The test suite

The resulting test suite consists of 305 test cases, split into 47 equivalence classes covering all of the feature types listed in the Materials and Methods section. 188 of the test cases are canonical forms and 117 are non-canonical[1]. The full test suite is freely available at

[1]This suggests that the test suite needs many more non-canonical forms. If we think of canonical forms as "clean" tests (ones which should be expected to pass) and non-canonical forms as "dirty" tests (ones which exercise a system's ability to handle unexpected inputs), "immature" testing organizations have been found to have a 5:1 ratio of clean:dirty tests, while "mature" ones have a 5:1 ratio of dirty:clean tests.

http://bionlp-corpora.sourceforge.net/testsuite/index.shtml.

3.2. Performance of the concept recognition system

- Accuracy on all canonical forms: 97.9% of all canonical forms were recognized. The missed canonical forms all contained the word *in*.
- Accuracy on all non-canonical forms: Non-canonical forms were not recognized at all.

3.2.1. Specific error types that immediately became apparent

A number of specific error types immediately became apparent. They fell into two broad families. The first family of errors is that the default system only recognizes terms when they appear in free text as exact matches of the term as it appears in the ontology. Important variant forms of the terms are not recognized. These include:

- Simple syntactic variants, such as *apoptosis induction* instead of the canonical form *induction of apoptosis* (GO:0006917)
- Coordinated forms
- Forms with any inserted text
- Simple morphological variants, including plurals
- Part of speech variants, including verb/noun substitutions (e.g. *cell migrates* instead of the canonical form *cell migration* (GO:0016477)) and adjective/noun substitutions (e.g. *nuclear* instead of the canonical form *nucleus* (GO:0005634))

The other family of errors concerned terms that contain the word *in*. The application was frequently found to fail to recognize terms that contain the word *in*, even when the terms appear in their canonical form. We have not yet been able to isolate the conditions under which this does and does not happen.

4. Discussion

Using our test suite, we were able to immediately identify a number of errors in the performance of an ontology concept recognition system, without the necessity of performing a search of the corpus for relevant cases. As (Oepen et al., 1998) point out, there are a number of advantages to using a test suite rather than naturally occurring corpora for the specific task of testing natural language processing software:

- Control over test data: Test suites allow for "focussed and fine-grained analysis of system performance" (15).
- Systematic coverage: Test suites allow for systematic evaluation of variations in a particular feature of interest.
- Control of redundancy: Test suites allow for reduction of redundancy when it obscures the situation, or for increasing it when it is important to test the handling of a feature whose importance is greater than its frequency in naturally occurring data.

To this we can add speed of execution—(Cohen et al., 2008) compared the efficacy of a structured test suite and a very large corpus in achieving code coverage (a measure of test sufficiency) and found that the test suite achieved higher coverage, even though the test suite took only eleven seconds to run and the corpus took four hours and 28 minutes.

The alternative to using a structured test suite is to use a corpus, and then search through it for the relevant inputs and hope that they are actually attested.

Can we find commonalities between (Cohen et al., 2004)'s equivalence classes and dimensions of variability that can lead us towards a general description of features that are likely to be relevant to any semantic class of biomedical named entity and to the very different application classes of gene mention systems and ontology concept recognition systems? Cohen et al.'s feature set comprised four broad categories: orthographic/typographic features, morphosyntactic features, source features, and lexical features. We have organized our dimensions of variability into inherent properties of terms (e.g. length and including punctuation) and properties of variable forms of the terms (such as pluralization and ordering). The level of granularity of the two approaches is quite different, and features in this work are divided amongst multiple categories in the (Cohen et al., 2004) typology, making direct comparison at the category level difficult. However, it is possible to compare individual dimensions of variability between the two typologies. Table 1 shows a tabular comparison between the two typologies.

We see that between the two systems, there are twenty features in total. Of these, seven are shared between the two models. This suggests that there is no single set of features that encompasses the needs of both types of systems, but is consistent with the hypothesis that there is a core of feature types that may be applicable to test suite construction for any similar type of application. Based on our comparison of the two feature sets, a first approximation of the core feature set would be:

- Length
- Numerals
- Punctuation
- Function/stopwords

[2]The parts of speech feature of the GM test suite, which has no correlate in the ontology concept recognition features, was never actually implemented in the GM dynamic test suite generator.

[3]The lexicographic feature of the GM test suite, which has no correlate in the ontology concept recognition features, was never actually implemented in the GM dynamic test suite generator.

[4]Syntactic context in the GM test suite work is separate from the four categories of features that we discussed above and includes much more than coordination.

GM features	Ontology concept recognition features
length	length
case	no correlate
numeral-related features	presence of numerals
punctuation-related features	punctuation
Greek-letter-related features	no correlate
function words	stopwords
parts of speech[2]	no correlate
source or authority	trivially present
original form in source	canonical terms
lexicographic features[3]	no correlate
no correlate	ungrammatical terms
no correlate	official synonyms
no correlate	ambiguous terms
no correlate	singular/plural
no correlate	ordering/syntactic variants
no correlate	inserted text
syntactic context[4]	coordination
no correlate	overlapping terms
no correlate	verbal vs. nominal
no correlate	adjectival vs. nominal

Table 1: Equivalence classes and dimensions of variability in (Cohen et al., 2004)'s gene mention system test suite and the ontology concept recognition test suite

- Source or authority
- Canonical form in source
- Syntactic context

Most of these features define equivalence classes, but we might also ask whether or not they give us insight into boundary conditions. Length and punctuation have obvious boundary conditions, but most of these classes do not. We have suggested in (Cohen et al., 2004) and here that designing test suites of this sort is helped not just by the principles of software engineering, but by the approach of descriptive linguistic field work. We note that except for the single feature that is essentially "bookkeeping," i.e. tracking the source or authority, all of the remaining six features are linguistically motivated. Length is of well-known linguistic importance, as mentioned above. Numerals and punctuation are written-language elements of morphology, and function words are a linguistic category. Canonical form is equivalent to the underlying form in any derivational theory of linguistics, and syntactic context is of course purely linguistically defined.

4.1. Future work

- In the current version, all terms are isolated, and not in sentential context, so it may not be useful for learning-based systems. Learning-based systems for ontology concept recognition are almost nonexistent at this time, so we feel comfortable leaving this for a future release.
- The current version only covers a single ontology, although it is the canonical biomedical ontology.
- We made no attempt to systematically sample the three sub-ontologies of the Gene Ontology.
- The ratio of "dirty" tests to "clean" tests should be increased.

5. Conclusion

Using our test suite, we were able to immediately identify a number of errors in the performance of an ontology concept recognition system, without the necessity of performing a search of the corpus for relevant cases. More broadly, we were able to find some core features for the design of this type of test suite.

6. References

Boris Beizer. 1990. *Software testing techniques, 2nd edition*. International Thomson Computer Press.

Boris Beizer. 1995. *Black-box testing: Techniques for functional testing of software and systems*. Wiley.

Douglas Biber, Stig Johansson, Geoffrey Leech, Susan Conrad, and Edward Finegan. 1999. *Longman grammar of spoken and written English*. Pearson.

Robert V. Binder. 1999. *Testing Object-Oriented Systems: Models, Patterns, and Tools*. Addison-Wesley Professional, October.

Christian Blaschke, Eduardo A. Leon, Martin Krallinger, and Alfonso Valencia. 2005. Evaluation of BioCreative assessment of task 2. *BMC Bioinformatics*, 6 Suppl 1.

Leonard Bloomfield. 1933. *Language*. University of Chicago Press.

E. B. Camon, D. G. Barrell, E. C. Dimmer, V. Lee, M. Magrane, J. Maslen, D. Binns, and R. Apweiler. 2005. An evaluation of go annotation retrieval for biocreative and goa. *BMC Bioinformatics*, 6 Suppl 1.

Werner Ceusters, Barry Smith, and Louis Goldberg. 2005. A terminological and ontological analysis of the NCI Thesaurus. *Methods of Information in Medicine*, 44:498–507.

K. Bretonnel Cohen, Lorraine Tanabe, Shuhei Kinoshita, and Lawrence Hunter. 2004. A resource for constructing customized test suites for molecular biology entity identification systems. In *HLT-NAACL 2004 Workshop: BioLINK 2004, Linking Biological Literature, Ontologies and Databases*, pages 1–8. Association for Computational Linguistics.

K. Bretonnel Cohen, William A. Baumgartner Jr., and Lawrence Hunter. 2008. Software testing and the naturally occurring data assumption in natural language processing. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, pages 23–30, Columbus, Ohio, June. Association for Computational Linguistics.

Kirill Degtyarenko. 2003. Chemical vocabularies and ontologies for bioinformatics. In *Proc 2003 Itnl Chem Info Conf*.

H.A. Gleason Jr. 1961. *An introduction to descriptive linguistics*. Holt, Rinehart and Wilson, revised edition.

Zellig S. Harris. 1951. *Methods in structural linguistics*. University of Chicago Press.

Lynette Hirschman, Marc Colosimo, Alexander Morgan, and Alexander Yeh. 2005. Overview of BioCreative Task 1B: normalized gene lists. *BMC Bioinformatics*, 6 Suppl 1.

Helen L. Johnson, K. B. Cohen, William A. Baumgartner, Zhiyong Lu, Michael Bada, Todd Kester, Hyunmin Kim, and Lawrence Hunter. 2006. Evaluation of lexical methods for detecting relationships between concepts from multiple ontologies. *Pac Symp Biocomput*, pages 28–39.

Helen L. Johnson, K. Bretonnel Cohen, and Lawrence Hunter. 2007. A fault model for ontology mapping, alignment, and linking systems. In *Pacific Symposium on Biocomputing*, pages 233–244. World Scientific Publishing Company.

Cem Kaner, Hung Quoc Nguyen, and Jack Falk. 1999. *Testing computer software, 2nd edition*. John Wiley and Sons.

Cem Kaner, James Bach, and Bret Pettichord. 2002. *Lessons learned in software testing: a context-driven approach*. John Wiley and Sons, Inc.

S. Kinoshita, K. B. Cohen, P. V. Ogren, and L. Hunter. 2005. BioCreAtIvE Task1A: entity identification with a stochastic tagger. *BMC Bioinformatics*, 6 Suppl 1.

Martin Krallinger, Florian Leitner, Carlos Rodriguez-Penagos, and Alfonso Valencia. 2008. Overview of the protein-protein interaction annotation extraction task of biocreative ii. *Genome Biology*, 9(Suppl. 2).

Brian Marick. 1997. *The craft of software testing: subsystem testing including object-based and object-oriented testing*. Prentice Hall.

Alexander A. Morgan, Benjamin Wellner, Jeffrey B. Colombe, Robert Arens, Marc E. Colosimo, and Lynette Hirschman. 2007. Evaluating human gene and protein mention normalization to unique identifiers. In *Pacific Symposium on Biocomputing*, pages 281–291.

Alexander A. Morgan, 17 other people, K. Bretonnel Cohen, and Lynette Hirschman. 2008. Overview of BioCreative II gene normalization. *Genome Biology*, 9.

Glenford Myers. 1979. *The art of software testing*. John Wiley and Sons.

S. Oepen, K. Netter, and J. Klein. 1998. TSNLP - test suites for natural language processing. In John Nerbonne, editor, *Linguistic Databases*, chapter 2, pages 13–36. CSLI Publications.

Ron Patton. 2005. *Software testing*. Sams, 2nd edition.

Randolph Quirk, Sidney Greenbaum, Geoffrey Leech, and Jan Svartvik. 1985. *A comprehensive grammar of the English language*. Longman.

William J. Samarin. 1967. *Field linguistics: A guide to linguistic field work*. Irvington.

Nigam H. Shah, Nipun Bhatia, Clement Jonquet, Daniel Rubin, Annie P. Chiang, and Mark A. Musen. 2009. Comparison of concept recognizers for building the open biomedical annotator. *BMC Bioinformatics*, 10.

C. Verspoor, C. Joslyn, and G. Papcun. 2003. The Gene Ontology as a source of lexical semantic knowledge for a biological natural language processing application. In *Proceedings of the SIGIR'03 Workshop on Text Analysis and Search for Bioinformatics*, Toronto, CA, August.

Martin Volk. 1998. Markup of a test suite with SGML. In John Nerbonne, editor, *Linguistic databases*, pages 59–76. CSLI Publications.

A. Yeh, A. Morgan, M. Colosimo, and L. Hirschman. 2005. BioCreatve task 1A: gene mention finding evaluation. *BMC Bioinformatics*, 6(Suppl. 1).