

Data and text mining

High-performance gene name normalization with GENo

Joachim Wermter*, Katrin Tomanek and Udo Hahn

Jena University Language and Information Engineering (JULIE) Lab, Friedrich-Schiller-Universität Jena, Fürstengraben 30, 07743 Jena, Germany

Received on December 11, 2008; revised on January 21, 2009; accepted on January 28, 2009

Advance Access publication February 2, 2009

Associate Editor: Jonathan Wren

ABSTRACT

Motivation: The recognition and normalization of textual mentions of gene and protein names is both particularly important and challenging. Its importance lies in the fact that they constitute the crucial conceptual entities in biomedicine. Their recognition and normalization remains a challenging task because of widespread gene name ambiguities within species, across species, with common English words and with medical sublanguage terms.

Results: We present GENo, a highly competitive system for gene name normalization, which obtains an F-measure performance of 86.4% (precision: 87.8%, recall: 85.0%) on the BIOCREATIVE-II test set, thus being on a par with the best system on that task. Our system tackles the complex gene normalization problem by employing a carefully crafted suite of symbolic and statistical methods, and by fully relying on publicly available software and data resources, including extensive background knowledge based on semantic profiling. A major goal of our work is to present GENo's architecture in a lucid and perspicuous way to pave the way to full reproducibility of our results.

Availability: GENo, including its underlying resources, will be available from www.julielab.de. It is also currently deployed in the Semedico search engine at www.semedico.org.

Contact: joachim.wermter@uni-jena.de

1 INTRODUCTION

Biomedical text mining deals with the challenge to automatically locate information contained in the life sciences literature in a way which is faster and more reliable than manual inspection or general-purpose search engines such as GOOGLE. One crucial step towards this goal is the automatic recognition of so-called named entities, e.g. names of genes and proteins, and their subsequent normalization and mapping to database identifiers. On the one hand, such normalization facilitates the integration of different knowledge sources, viz. unstructured textual and structured database information. On the other hand, from an information retrieval perspective, entity normalization substantially eases indexing by a search engine and querying through its user interface.

The normalization of gene mentions in plain text is both particularly important and challenging. It is important because both genes and proteins constitute the crucial conceptual entities in biomedicine, as is witnessed by the respective tasks of the BIOCREATIVE-I (Hirschman *et al.*, 2005) and BIOCREATIVE-II

(Hirschman *et al.*, 2007) competitions. At the same time, despite many efforts it remains a challenging task because of widespread gene name ambiguities within species, across species, with common English words and with medical sublanguage terms (Chen *et al.*, 2005). In general, gene mapping can be decomposed into several subtasks that all need to be addressed to come full circle:

1. *Gene Mention Detection:* this first step basically involves named entity (NE) recognition, i.e. the detection of named entities which denote gene names. The entities then serve as candidates with which a gene database identifier may be associated.
2. *Building a Gene Name Dictionary:* since there is no single comprehensive and well-curated lexical resource for gene names, gene dictionaries have to be compiled from various protein or gene databases, e.g. UNIPROT (<http://www.uniprot.org>) or ENTREZGENE (<http://www.ncbi.nlm.nih.gov/sites/entrez?db=gene>).
3. *Tokenization/Variant Generation:* even carefully compiled and up-to-date gene dictionaries typically do not contain all possible synonyms or name variants which can be found in text. A common way to address this shortcoming is automatic term variant generation guided by plausible naming rules or tokenization procedures which lead to canonical token forms.
4. *String Matching:* in the mapping step, the gene mentions found in text need to be linked with their corresponding gene dictionary entries. Despite lots of efforts in devising apt variant generation or tokenization procedures which are intended to lay the ground for exact matches, approximate string matching is still indispensable due to proliferating term variability.
5. *False Positive (FP) Filter:* after one or several gene dictionary matches have been found for a gene mention, a gene name normalizer decides whether to keep a match (and to assign a database identifier to it), or to discard it. The latter is a crucial step in case the match is a FP (e.g. a cell or disease name) or more than one match is possible, i.e. gene name ambiguity (e.g. 'p21' may refer to several gene identifiers).

While a lot of studies were run on each of these subtasks in isolation, there are only few approaches that propose an integrated architecture for gene mapping in a comprehensive real-world biomedical text mining system. This number gets even smaller when the requirement is added that the system actually show a competitive performance on community-wide accepted datasets such as BIOCREATIVE. Remarkably, these rare systems,

*To whom correspondence should be addressed.

in turn, are either characterized by the need for continuous and laborious manual dictionary curation (Hanisch *et al.*, 2005), or by a considerable degree of opaqueness with respect to methodological and implementational issues (Hakenberg *et al.*, 2008b), or by both.

To address these shortcomings, we developed GENO, a fully integrated system for gene name normalization. It is built from open-source libraries and publicly available resources only, offers a lucid and perspicuous architecture with emphasis on ease of understanding and the potential for reproducibility, and achieves state-of-the-art results on the BIOCREATIVE-II gene normalization dataset. Methodologically and experimentally, we show that (i) (ML) methods perform superiorly when integrated with publicly available training data in a well-designed manner and (ii) a simple bag-of-words semantic approach to biological background knowledge performs as well as more complex semantic disambiguation computations [e.g. as in Hakenberg *et al.* (2008a)].

Besides considering already existing approaches and systems in Section 2, this article is organized as follows: Section 3 describes in detail the architecture and components, as well as the resources employed in the GENO system. In Section 4, we deal with the experimental settings in which individual components were run and evaluated in different modes on the BIOCREATIVE-II dataset. Section 5 contains an extensive discussion of the results obtained.

2 RELATED WORK

Finding gene mentions in the biomedical literature and normalizing these names to their respective database identifiers is a well-established research problem. While many publications exclusively deal with specific subtasks, only few papers cover the whole task.

Tsuruoka *et al.* (2007) focus on matching putatively related strings, i.e. variants and synonyms of a gene name. They find that an ML approach, where a maximum entropy model predicts a similarity score for pairs of gene mentions, outperforms simpler string-based approaches. The training data are taken from existing gene dictionaries which, however, do not necessarily reflect gene mentions as they occur in real text. The recall-only evaluation is performed either dictionary-internally, or on the already given gene mentions of the BIOCREATIVE-II test set.

Xu *et al.* (2007) examine the alternative subtask of gene name disambiguation under several idealistic assumptions. First, perfect gene mentions are assumed most of which are restricted to short-string gene symbols. Another unrealistic condition they introduce is that among the possible gene candidates in their disambiguation task one candidate is always the correct answer. This assumption ignores the fact that an apparent gene mention in text may not denote a gene at all (i.e. a FP is encountered). Furthermore, their knowledge-based profiling approach compares several semantic profiling techniques (some of them being quite complex). A look at their results, however, reveals that a plain bag-of-words approach performs almost equally well [cf. Table 2 in Xu *et al.* (2007)].

PROMINER (Hanisch *et al.*, 2005) is a well-known and complete gene name normalization system. Employing a strict dictionary-based approach, it heavily relies on the (manual) curation and quality of its gene dictionaries. Disambiguation is achieved by finding other synonyms of ambiguous gene names in the same text. However, because of PROMINER's proprietary nature, many methodological and implementational issues remain hidden.

GNAT (Hakenberg *et al.*, 2008b) is a comprehensive inter-species gene name normalizer building on the winner system of the BIOCREATIVE-II competition as described in Hakenberg *et al.* (2008a). For gene mention detection and mapping, it employs dictionary- and ML-based modules though its dictionary component appears to be dominant in the processing cycle (relying on separate dictionaries for 25 species). Permissive generation rules which are compiled into finite-state automata account for gene name variants, while gene name disambiguation is performed using several sources of background information (ENTREZ GENE summaries, GO annotations, several UNIPROT fields, etc.), as well as complex similarity scoring.

3 METHODS

As discussed in Section 1, gene name normalization consists of several subtasks including gene mention detection, construction of gene name dictionaries, tokenization or variant generation, string matching and finally filters to discard FPs and ambiguities. This section describes in detail the GENO approach to gene name normalization, its architecture and underlying methods. Furthermore, we outline how essential resources—including, e.g. training material and dictionaries—incorporated in GENO were (semi-)automatically compiled from publicly available repositories.

3.1 Architecture and components

Figure 1 shows the overall architecture of GENO. Once gene mentions have been identified, such mentions are tokenized relying on special rules and then fed into a retrieval procedure which returns an ordered list of candidate identifiers for each gene mention. These candidates are then handed over to an assessment component which, based on semantic similarity, decides which (and whether) one of the candidates is the correct gene name identifier.

3.1.1 Gene mention detection Dictionary-based approaches to gene mention detection usually employ already pre-compiled lexical resources. It is well-known that due to the variability of biomedical names, such resources tend to be incomplete in the sense that they do not enumerate all possible ways a particular gene name may surface in text. Hence, such lexical resources are often extended by manual curation efforts, by (semi-)automatic

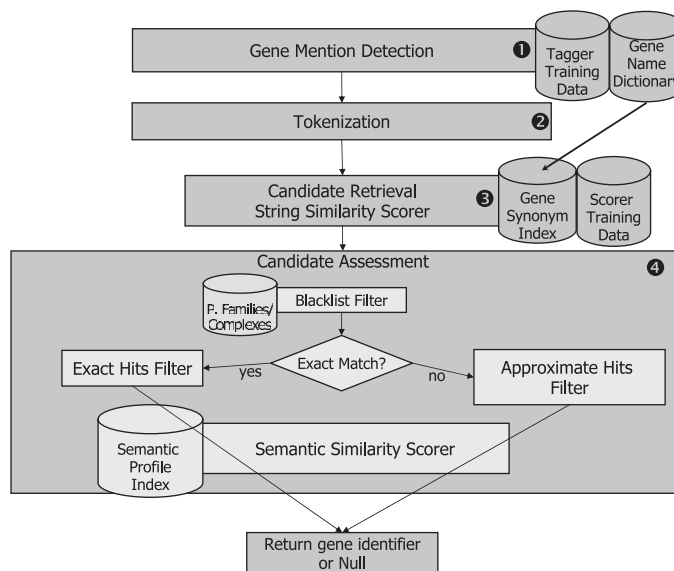


Fig. 1. Architecture of the GENO gene name normalization system.

means (Hakenberg, 2007), or by a mixture of both (Hanisch *et al.*, 2005). Unfortunately, dictionary-based gene mention detection typically recognizes more FPs because dictionaries often contain entries that overlap with other biomedical terms or common English lexical items.

NE recognition based on supervised ML has also been successfully applied to gene mention detection (Leaman and Gonzalez, 2008; Settles, 2004). Unlike dictionary-based approaches, ML-based NE taggers are good at generalizing to new data and are thus able to recognize gene names not seen during training. While often suffering from recall lower than dictionary-based approaches, ML methods are typically superior in terms of precision since less FPs are tagged. Still, ML-based gene mention detection does not come for free because considerable manpower is required to annotate high-quality training data.

Although several gene name-annotated corpora are already available, they tend to be rather specialized, i.e. either constrained to certain organisms or to specific biomedical topics. For example, the GENIA corpus (Kim *et al.*, 2003) is focused on a subset of human hematology, whereas the PENNBIOIE corpus (Kulick *et al.*, 2004) is on oncology. Consequently, the kinds of genes occurring in these corpora are rather specialized and high-performing ML-based gene taggers trained on them may miss many gene names in texts from a different subdomain. Hence, a proper gene name-annotated training corpus should cover a wide range of biomedical domains. Section 3.2.1 discusses how we compiled such high-coverage training material out of publicly available gene name annotated corpora.

As the ML-based gene tagger of choice, we here employ JNET, the JULIE Lab Named Entity Tagger (Hahn *et al.*, 2008). This tagger is based on Conditional Random Fields (Lafferty *et al.*, 2001) and incorporates features which have been shown to work well for biomedical entity recognition (Settles, 2004). In order to address possible lower recall, we integrate as a complementary methodology dictionary-based gene name detection to find gene mentions that were missed by the ML-based gene tagger. Section 3.2.2 discusses how we compiled the underlying dictionary.¹

3.1.2 Tokenization A common problem with dictionary-based approaches is their blindness with respect to unseen gene names and their lack of flexibility to account for term variations of listed gene names. One way to address the latter issue is to generate lots of variants by exploiting structural, lexical, orthographical or morphological properties of names (Hakenberg, 2007; Hakenberg *et al.*, 2008b). Augmenting a dictionary with additional entries in such a way, however, suffers from an inherently combinatorial growth behavior of the variant set.

To circumvent this unwarranted effect, we made attempts at ‘reverse engineering’ of token variants. This technique also exploits orthographical properties of gene names, yet in a reductive manner in that it splits up non-canonical character sequences (lower case followed by upper case, alphabetical followed by numerical characters and vice versa, etc.) and deletes special characters and stop words to obtain one single normalized name form for all its variants. For example, the gene mentions ‘*sIL-1 Beta2*’, ‘*sIL-1BETA-2*’, ‘*sIL1-beta-2*’, ‘*sIL-1 beta2*’ and ‘*sIL1beta2*’ will all be tokenized to ‘*s il 1 beta 2*’. Such a reductive tokenization has similar effects as variant generation without running into combinatorial problems. Our reductive tokenization rules are both applied to the gene mentions found in text, as well as to those appearing in our gene dictionary.

3.1.3 Candidate Retrieval All tokenized gene mentions are matched against the entries in our gene dictionary to find potential identifiers. We employ the indexing and retrieval facilities of the APACHE LUCENE (<http://lucene.apache.org>) search engine for efficient candidate retrieval. From an implementational point of view, each synonym of our *Gene Name Dictionary* is stored as a ‘document’ in the LUCENE-based *Gene Synonym*

Index. Tokenized gene mentions found in text are queried against this index which, in turn, yields a ranked list of gene dictionary entries matching this query. We are using Boolean queries so that the actual word order within the gene mentions does not play any role. This is advantageous as often word order permutations occur when, e.g. prepositions are used. Queries for gene mentions, such as ‘*receptor of IL 2*’ and ‘*IL 2 receptor*’, thus lead to full matches since our tokenization also removes ‘*of*’ as a stop word.

The results of such queries are re-ranked by our *String Similarity Scorer* which aims at determining the similarity between the candidate term from the dictionary and the gene mention in text on a surface string level. While we could use the already LUCENE-computed term frequency/inverse document frequency (TF-IDF) score, we instead employ an ML-based approach which is similar to the one proposed by Tsuruoka *et al.* (2007). In concrete, a Maximum Entropy model is used to score surface string similarity between two gene names.² The features used for this scorer indicate whether or not two names share the same tokens, the same character bi-grams, the same prefixes and suffixes, etc. We augmented these rather generic features with gene name-specific ones, such as whether or not two names share the same molecular weight (e.g. ‘*p65*’), the same Greek letter (e.g. ‘*beta*’), or the same gene name specifier (e.g. ‘*receptor*’, ‘*ligand*’). In Section 4, we show that our ML-based scorer indeed outperforms LUCENE’s default score. However, this is only true when real-world training material is used (Section 3.2.3).

3.1.4 Candidate Assessment Once candidates have been identified, we still need to assess their appropriateness on a semantic level. This is realized by several string matching and scoring steps, so-called filters. The result of this filtering is always exactly one hit or NIL. Our first filter pays tribute to the BIOCREATIVE data which exclude any mentions of protein families, groups or complexes. For this reason, a *Blacklist Filter* aims at eliminating such gene mentions if they match an entry on a respective blacklist. Section 3.2.4 explains how this list was compiled.

Even if the gene mention and the candidate gene name are exactly the same, it is still necessary to assess the quality of such exact hits. One reason is that perfect matches may turn out to be target FPs in that they do not refer to gene names at all but rather to disease names, cell names, etc. Another reason is ambiguity, i.e. more than one exact match is possible. Our *Exact Hits Filter* is based on a *Semantic Similarity Scorer* (see Section 3.1.5 for details) which estimates the semantic similarity between the document in which the gene mention was found and the semantic profile of the putative gene hit. The exact match with the highest semantic score is chosen as the correct hit—but only if it exceeds a certain threshold. Thus, the *Semantic Similarity Scorer* has both a disambiguation function (in case of more than one exact hit) and a FP filter function.

Despite reductive tokenization, there will be gene mentions, in particular those found by the ML-based gene tagger, which may not be matched exactly with a dictionary entry. We then apply our *Approximate Hits Filter*. In a preliminary filtering step, all candidates are discarded for which the overlap with the gene mention only consists of single characters or digits or various gene name-signaling keyword classes, such as as Greek letters, modifiers, specifiers or non-descriptives.³ This is due to the fact that our tokenization splits many gene names into numerous subtokens which according to the transformations being made erroneously seem to indicate surface string similarity although this is not backed up from a lexically semantic point of view. The remaining candidates are then also ranked by the *Semantic Similarity Scorer*. *Thresholding* (Section 3.1.6) is employed and the candidate with the highest semantic score is returned as the correct gene identifier. If no candidate scores above the threshold, no mapping for this gene mention is recorded.

3.1.5 Semantic Similarity Scorer For exact as well as approximate matches, a semantic score is calculated to check whether the candidate and

¹For dictionary look-up, we used LINGPIPE’s dictionary chunker (<http://alias-i.com/lingpipe>) and the dictionary annotator from UIMA’s sandbox (<http://incubator.apache.org/uima/sandbox.html>).

²We employed the MALLET ML package (http://mallet.cs.umass.edu/index.php/Main_Page) for this purpose.

³See Hanisch *et al.* (2005) for an extensive description of these classes.

the actual gene mention are semantically similar. The *semantic similarity scorer* uses a *Semantic Profile Index* also built upon LUCENE's retrieval facilities. This index contains the semantic profile of each dictionary gene identifier and was compiled as a (stemmed and stop word-removed) bag of words from various gene/protein databases and ontologies (for details, see Section 3.2.5). For the current version of the GENO system, semantic similarity is computed by querying the whole abstract (stemmed and stop word-removed bag-of-words) text in which a particular gene mention occurs against the *Semantic Profile Index*. A list ranked by LUCENE's built-in TF-IDF-based score is then returned: genes whose semantic profiles are more similar according to the TF-IDF score are ranked higher.

3.1.6 Thresholding Calibrating thresholds is a delicate issue (not only) for gene name normalization systems. We need to set thresholds for exact and approximate hits returned by the *Semantic Similarity Scorer*. Our experiments have shown that the thresholds for approximate hits should be higher than for exact ones. This makes sense in as much as exact hits may be seen as a 'safer bet' than approximate ones.

For gene mentions found by the dictionary-based tagger, we only deal with exact hits in the first place (since exact dictionary look-up is performed). As our experiments show, dictionary tagger-derived gene mentions may not be trusted as much as ML tagger-derived (exact) gene mentions. This is particularly true, as in our case, where the gene dictionary has not undergone an extensive curation process and thus still may contain numerous noisy entries. In such a case, the semantic threshold has to be set higher, in particular, if the dictionary-based gene tagger is used in combination with the ML-based one. However, in the (experimental) case in which we only wanted to use the dictionary-based tagger alone, the semantic threshold may not be set as high (see Section 4.2 for an extensive discussion on this).

3.2 Compilation of resources

The resources we used for gene mapping are all freely available. However, pre-processing is mostly necessary to obtain suitable resources that can be used by our components. All these modified resources underlying our experiments will be available from www.julielab.de.

3.2.1 Training Data for the ML-based NE tagger We collected various publicly available gene-annotated corpora and converted their annotations for training purposes into the common IO scheme in which each word is either marked as belonging to a gene name (I—inside) or not (O—outside). Based on the gene annotations from GENIA (Kim *et al.*, 2003), PENNBIIE (Kulick *et al.*, 2004), GENETAG (Tanabe *et al.*, 2005), PIR (Mani *et al.*, 2005) and AiMED (Bunescu *et al.*, 2005), this yielded a gene-annotated corpus of about 2 million tokens, which is rather heterogeneous and noisy—not only relative to the subdomains covered, but also reflecting different word tokenization schemes and gene name annotation guidelines.⁴ To evaluate the performance of our ML-based classifier on this merged corpus, we performed a 10-fold cross-validation on the data. The tagger achieved 80.1 on balanced F-score, with precision and recall peaking at 79.8 and 80.4, respectively.⁵

3.2.2 Gene Name Dictionary The most comprehensive and up-to-date lexical resources for gene names may be harvested either out of organism-specific genomic databases (such as MGI) (<http://www.informatics.jax.org>) or out of centralized resources, such as UNIPROT or ENTREZGENE which are also partly fed from organism-specific databases. There have also been

efforts to compile gene names from the major existing databases into a single lexical resource, e.g. the BIOThESAUrus (BT) project (Liu *et al.*, 2006).

The main purpose of a gene name dictionary in our work is to provide the actual database identifiers to be assigned to the gene mentions in text. Although BT would be a natural choice in a practical system, its UNIPROT perspective on gene identifiers introduces a difficulty with respect to system evaluation on the standard BIOCREATIVE-II dataset which is ENTREZGENE-centered. Accordingly, there is not always a direct one-to-one mapping between the two sets of database identifiers. As a consequence, for evaluation purposes, we mainly stuck to the original gene dictionary provided by the BIOCREATIVE-II organizers⁶ although we enhanced it with updated entries from the UNIPROT database. This enhanced dictionary was used both as the basic resource for the dictionary-based gene mention detection, as well as to create the LUCENE-based *Gene Synonym Index* used for *Candidate Retrieval*.

3.2.3 Training Data for String Similarity Scorer Concerning our ML-based *String Similarity Scorer* used to rank approximate matches as described in Section 3.1.3, Tsuruoka *et al.* (2007) claim that such an approach may outperform standard statistical measures (such as TF-IDF or Jaro-Winkler) with respect to recall. In concrete, Tsuruoka *et al.* collected positive and negative training examples pairwise by selecting two gene names which share surface string properties (e.g. 'il 2 receptor gamma' and 'interleukin 2 receptor gamma'). They thus qualify as either *positive pairs* (in case they refer to the same gene identifier) or *negative pairs* (in case they do not, as with 'il 2 receptor alpha' and 'il 2 receptor gamma').

The experiments conducted by (Tsuruoka *et al.*, 2007) were exclusively based on gene names taken from dictionaries, which may not reflect their actual use in real text. For this reason, we took an alternative avenue and collected training examples from real text data. This may ideally be done in such a way as to select one member of the pair to be the textual gene mention and the other one to be a candidate from the *Gene Name Dictionary* (i.e. the *Gene Synonym Index* in component 3 from Fig. 1). For this purpose, we used the BIOCREATIVE-II noisy training data (Morgan *et al.*, 2008) which contains about 4000 Medline abstracts with ENTREZGENE identifiers assigned to each (obtained through GENERIF and GOA annotations) on the document level. We then ran component 1 through 3 of the GENO system (cf. Fig. 1) on these abstracts (performing gene mention detection, tokenization and candidate retrieval), with component 3 employing LUCENE's built-in TF-IDF-based score for the *String Similarity Scorer* and returning a ranked list of gene identifiers. If there was a correct identifier (i.e. one that was assigned to the respective abstract) among the top 10 candidates, we used the associated dictionary gene name together with the textual gene mention (delivered by the gene tagger component) as a *positive pair*. The selection of *negative pairs* was done likewise for incorrect gene identifiers. Thus, we were able to train our Maximum Entropy-based ML *String Similarity Scorer* on real-text gene mentions paired with gene dictionary entries.

3.2.4 Blacklist Filtering Since the BIOCREATIVE-II annotation guidelines (Morgan *et al.*, 2008) on purpose excluded protein families, groups and complexes, we implemented a *Blacklist Filter* in the *Candidate Assessment* component (Section 3.1.4) which aims at discarding such unwarranted gene mentions. This list was compiled out of several publicly available resources. First, we consulted Wikipedia which contains several lists listing gene and protein families. For protein complexes, we added all terms stored under the MESH terms '*Biopolymers*' and '*Multiprotein Complexes*'. Furthermore, we augmented this list with protein family and complex annotations taken from the GENIA corpus.⁷

3.2.5 Semantic Profile Index The notion of semantic profiles of genes—sometimes called biological background knowledge—has recently been

⁴For example, whereas GENIA distinguishes between mentions at the DNA, RNA and protein level, as well as at the family, complex, molecule and subunit level, GENETAG only annotates actual gene mentions to which a database identifier may be assigned.

⁵The evaluation metric employed was strict in that exact boundary matches were required. Partial overlaps of gene mentions were discarded although for normalization purposes this softer condition may still be useful.

⁶Morgan *et al.* (2008) describe the compilation and cleansing of BIOCREATIVE-II's lexical resource.

⁷The GENIA corpus contains extensive annotations of gene/protein families, groups, complexes, domains and sites.

shown to hold much promise for the selection among different gene match candidates. There are a variety of resources, mostly biological databases and ontologies, from which such profiles can be constructed. For the *Semantic Similarity Scorer* module (Section 3.1.3), we constructed such a profile for each dictionary gene (identifier) using the ENTREZGENE summary (if available), the chromosome location for each gene, Gene Ontology (GO) Annotations from GOA, keywords assigned to each entry in the UNIPROT database and UNIPROT free-text field comments.⁸ Unlike Xu *et al.* (2007), we did not use the actual identifiers or codes from these sources (Go codes, etc.) but collected plain natural language descriptions and available synonyms (in the case of Go terms), thus constructing a ‘bag-of-words’ semantic index for each dictionary gene identifier.

4 EXPERIMENTS AND RESULTS

4.1 Experimental settings

For the evaluation of GENo, we varied different parameters of its constituent components to assess which modes were beneficial for the overall system. We focused on those components for which alternative parametrizations had already been discussed in the literature and, given our modular design, could easily be exchanged.

For *Gene Mention Detection*, we investigated whether the ML-based gene tagger, the dictionary-based one or a combination of both would yield the best performance results. For *Tokenization*, we examined whether the reductive gene name tokenization would pay off, or whether a more lenient tokenization—basically eliminating special characters as proposed by Tsuruoka *et al.* (2007)—would be sufficient. For the *String Similarity Scorer* used during the *Candidate Retrieval* phase, we tested whether an ML-based similarity model trained on real-world text data of gene names would fare better than a model derived only from gene dictionary-internal training—again, as advocated by Tsuruoka *et al.* (2007). We used the *Gene Name Dictionary* to derive appropriate dictionary-internal training data. In addition, we also checked the performance of LUCENE’s built-in TF-IDF-based scorer when used in exchange of the ML-based *String Similarity Scorer*. This is particularly interesting because LUCENE’s scorer comes ‘for free’ as the search engine calculates scores anyway.

Therefore (Table 1), as fundamental experimental conditions we distinguish between three different gene tagger modes: a pure ML tagger, a pure Dictionary-based tagger and a combination of both (M+D). For all three cases, a Reductive tokenization and a String Similarity Scorer using ML-based Real text data are applied. The three different experimental modes are named M–R–MR D–R–MR and M+D–R–MR, respectively. Since the combination of ML-based and dictionary-based NE tagging (M+D) yields the best results, all further tests employ this best-performing tagging mode.

The second round of experiments varies tokenization modes by contrasting Lenient tokenization (M+D–L–MR) with the Reductive one. Since lenient tokenization did worse than the reductive one, we focused on the latter in the final round of experiments. This third setting compares different String Similarity Scorers, viz. ML-based on Dictionary-internal data (M+D–R–MD) versus LUCENE’s TF-IDF scoring (M+D–R–LU). All experiments were run against the BIOCREATIVE-II test set on human genes. The thresholds, however, were calibrated on the BIOCREATIVE-II training set.

⁸The following fields were used: *protein function*, *sub-cellular location*, *tissue expression*, *interactions* and *disease*.

Table 1. Performance evaluation of GENo under various experimental conditions

Experimental condition			F1	Prec	Rec	TP	FP	FN
tag	tok	sim						
M+D	R	MR	86.4	87.8	85.0	668	93	118
M	R	MR	83.4	89.0	78.4	616	76	170
D	R	MR (i)	53.8	92.0	38.0	299	26	487
D	R	MR (ii)	79.5	83.4	76.0	597	119	189
M+D	L	MR	83.5	87.4	79.5	625	90	161
M+D	R	MD	83.8	88.6	78.9	620	60	166
M+D	R	LU	84.7	86.8	82.7	650	90	136

The experimental conditions are specified by a combination of shortcuts: the left part stands for the tagger type (M for ML-based, D for dictionary-based), the middle part specified the tokenization mode (R for reductive, L for lenient), and the right part characterizes the String Similarity Scorer chosen (MR for ML-based on real data, MD for ML-based on dictionary-based training data, and LU for LUCENE’s TF-IDF score).

4.2 Results

Table 1 summarizes the results for the different experimental settings. We employed the standard evaluation measures recall, precision and balanced *F*-score (*F*1), which were also used for the BIOCREATIVE-II competition. To better assess the performance differences between the different modes, Table 1 also displays the amount of true positives (TPs), FPs and false negatives (FNs). The best-performing combination of parameters, M+D–R–MR, employs both ML-based JNET and dictionary-based gene taggers, reductive tokenization and ML-based string similarity scoring trained on real text data. This mode achieves an overall *F*-score of 86.4 which outperforms any of the results reported for the BIOCREATIVE-II competition⁹ and equals the *F*1 results in a follow-up study reported in Hakenberg *et al.* (2008a), the best system on the gene normalization task in that competition.

4.2.1 Modes of gene mention detection If gene mention detection is only performed by the ML-based gene tagger (M–R–MR), the *F*-score performance drops by 3 points compared to the best performing scenario where both the ML-based and the dictionary-based tagger are used (M+D–R–MR). Compared with this superior setting, there is a substantial drop in the number of TPs from 668 to 616 (and thus a corresponding increase of FNs), while the number of FPs decreases less sharply by 17.

For the dictionary-based tagger (D–R–MR), we have to keep in mind that semantic thresholds (Section 3.1.6) for exact hits need to be set differently depending on whether the tagger is used in combination with the ML-based gene tagger (M+D–R–MR) or alone (D–R–MR). Thus, when the combination of both taggers is run, the threshold should be set higher than in the case when the dictionary-based tagger is used alone. Accordingly, we show two results for the D–R–MR mode. For the same (high) threshold as in the M+D–R–MR mode (i), the *F*-score performance drops drastically by more

⁹In terms of performance, we refrain from comparing ourselves directly to the systems which participated in the BIOCREATIVE-II challenge because these systems were developed under tightly scheduled circumstances.

than 30 points mainly due to a very low recall (38.0).¹⁰ After re-calibration for a more optimal threshold on the BIOCREATIVE-II training set (ii), the performance of the M+D–R–MR mode amounts to still unsatisfactory 79.5 *F*-score.

4.2.2 Modes of tokenization The results for the M+D–L–MR mode where reductive tokenization is replaced by a more lenient, special-character-based one, show that the performance drops by almost 3 points in terms of *F*-score to 83.5. As Table 1 reveals, this shift mostly occurs in terms of much lesser TPs being mapped (and thus resulting in more FNs). This is mainly due to the fact that during *Candidate Retrieval*, fewer gene candidates are retrieved from the *Gene Synonym Index* via the *String Similarity Scorer*. Considering the example from Section 3.1.2 again, the gene mention ‘*sIL1beta2*’ cannot be matched against ‘*IL1-beta-2*’ because lenient tokenization just replaces the special characters with whitespace (i.e. ‘*IL1 beta 2*’).

4.2.3 Modes of the String Similarity Scorer As for scoring, the string similarity between a gene mention yielded by one of the taggers and a candidate taken from the *Gene Synonym Index*, the best-performing combination of modes, M+D–R–MR, runs the *String Similarity Scorer* with an ML model derived from real text training data. Exchanging this model by one derived from artificial dictionary-internal training data (M+D–R–MD) results in a performance drop by 2.6 points *F*-score (cf. Table 1). Recall is particularly affected as it falls from 85.0 to 78.9, which is also reflected by the inverse shift from TPs to FNs. Interestingly, just keeping LUCENE’s pre-computed TF-IDF score as *String Similarity Scorer* when querying a gene mention against the *Gene Synonym Index* (M+D–R–LU mode) yields a better result (0.9 *F*-score up) than the M+D–R–MD mode. This shows that although an ML-based *String Similarity Scorer* can be beneficial, it is important to choose the right training material. Otherwise, standard metrics can do at least as well with considerably less overhead.

4.3 Error analysis

An analysis of FN and FP errors occurring for the best-performing scenario (M+D–R–MR) is given in Table 2. Errors were calculated by comparing the set of GENO’s mapped genes with the BIOCREATIVE-II gold standard on a per-document basis. In order to better understand the causes of errors, we categorized them such that they could be related to certain components or resources. Thirty-seven of the FN (31%) and 17 of the FP (18%) errors are due to the *Semantic Similarity Scorer* either by assigning a too low semantic similarity score (with respect to the calibrated threshold), or by assigning the best score to the wrong gene identifier. The *Gene Mention Detection* component is responsible for 20 of the FN errors (17%) by not or only partly tagging the gene mentions, and for 5 of the FP errors (5%) by tagging non-gene names erroneously. Twenty-five of the FN errors (21%) are due to deficiencies in the *Gene Name Dictionary*, with gene synonyms being too dissimilar or not being present at all. The *Blacklist* of protein families and complexes contains few wrong entries causing 5 FN errors, but more importantly, it still lacks coverage as it causes 35 of the FP errors

Table 2. Frequency of occurrence of FN and FP error types and the component/resource attributed (⊙ means none can be attributed)

Error type	Component/Resource	Freq.
False negatives		118
Semantic similarity score too low	Semantic Sim Scorer	37
Mention not or only partly tagged	Gene Mention Detect	20
Too dissimilar synonym in dictionary	Gene Name Dict	16
Missed coordination	Not yet available	15
No synonym in dictionary	Gene Name Dict	9
Due to FPs	⊙	5
Too aggressive filtering by blacklist	Blacklist	5
Miscellaneous cases	⊙	11
False positives		93
ID assigned to family/complex/domain	Blacklist	35
Wrong ID has highest semantic similarity	Semantic Sim Scorer	17
Gene refers to non-human organism	Not yet available	10
Gene is not annotated in gold standard	⊙	9
Name does not refer to a gene	Gene Mention Detect	5
Miscellaneous cases	⊙	17

(38%) which all refer to names for gene/protein families, complexes and domains/motifs.

Overall, the FN and FP errors indicate that our system could readily benefit from two additional components, in the first place. As discussed in Baumgartner *et al.* (2007), 8% of all gene names in BIOCREATIVE-II are part of conjunctions and ranges. The lack of a component in the GENO system to resolve such cases is reflected by 15 FN errors. On the FP error side, 10 errors are due to assigning gene identifiers to mentions from non-human organisms. Species-dependent normalization, although a complex task in itself, is thus eventually inevitable in a routine gene normalization system.

5 DISCUSSION AND CONCLUSIONS

GENO is a high-performance gene name normalization system which outperforms all but one state-of-the-art system. On human gene names on the BIOCREATIVE-II test set it peaks at an *F*-score of 86.4 in our best setting, which is identical with the performance figure reported by Hakenberg *et al.* (2008a) in a follow-up study to BIOCREATIVE-II. Unlike all these systems, GENO is entirely built upon publicly available resources and its performance can thus be fully reproduced and, possibly, adapted to new requirements. In addition, it does not require laborious manual or semi-automatic curation efforts of its lexical resources. In terms of efficiency, it typically takes less than a second to run a MEDLINE abstract through the system on dual quad-core LINUX server with 12 GB RAM.

Our results show that, among alternative methods available for crucial subtasks in a gene name normalization system, clear preferences take shape. With respect to the *Gene Mention Detection* task, the results indicate that a combined, bagging-style ML plus dictionary-based approach to gene mention tagging is superior to approaches that use only one or the other. It should be noted that the resources both for the ML-based tagger (i.e. the training corpora)

¹⁰Note that precision here is unusually high because the dictionary candidates that pass the (high) M+D threshold tend to be actual genes.

and for the dictionary tagger were not only derived from publicly available sources only, but were also virtually taken ‘as is’. This is particularly noticeable because several alternative approaches require heavy investments in manual resource curation (e.g. gene name dictionaries) in order to provide high-quality resources to feed their tagger Hanisch *et al.* (2005).

In what concerns the variability of gene names, our results indicate that a reductive *Tokenization* scheme (based on substring elimination and radical segmentation) should be favored over a more lenient one. We did not consider the other major alternative to tokenization discussed in the literature, viz. variant generation Hakenberg (2007). Rather than methodological issues, we see complexity and maintainability issues at stake here, especially if variant generation rules are overly permissive as advocated by (Hakenberg *et al.*, 2008b).

Complementing findings reported, e.g. by (Tsuruoka *et al.*, 2007) our results show that if an ML-based *String Similarity Scorer* is used for approximate matches, deriving a model from real-text training data clearly outperforms a scorer for which the model was obtained from artificial gene dictionary-internal data only. In fact, this latter approach is even outperformed by the readily available score the LUCENE search engine assigns to each candidate, by default.

Our error analysis indicates that there is room for improvement with respect to the *Semantic Similarity Scorer* which, up until now, uses a global bag-of-words scoring mechanism comparing the whole abstract text with the bag-of-words *Semantic Profile Index* of each candidate identifier. It would be worthwhile to test whether smaller, more local, units (e.g. the sentence in which a gene name occurs) yielded more adequate semantic similarity scores so that the number of FN errors could be reduced. Still, it should be noted that our straightforward global approach, in combination with the other components, already outperforms more sophisticated semantic similarity and disambiguation schemes presented, e.g. by Hakenberg *et al.* (2008b) or Xu *et al.* (2007). This argument even holds in direct comparison with the on-par performance results reported in Hakenberg *et al.* (2008a) because simplicity, perspicuity and ease of use and adaptation are major scientific preference criteria to choose among competing approaches.

Another interesting result in our error analysis shows that almost 13% of the FN errors are due to missed coordinations (i.e. syntactic constructions such as ‘*SMADs 1, 5 and 8*’). The best performing system on the BIOCREATIVE-II dataset (Hakenberg *et al.*, 2008b) already includes components to handle this task and thus a similar module in GENo might boost its performance as well.

Furthermore, 9% of the FP errors can be traced to the lack of a module in GENo which associates gene mentions with a particular organism. However, adding such a module may not necessarily improve the performance because inter-organism disambiguation is a complex task and the false assignment of a particular organism to a gene mention leads to both a FP and a FN. Still, such a module is essential for a real-world system and thus is high on our agenda.

Funding: German Ministry of Science and Education (BMBF) (STEMNET project, Funding-ID: 01DS001B partial); European Commission within the 6th Framework Programme (BOOTSTREP project, Contract: 28099, partial).

Conflicts of Interest: none declared.

REFERENCES

- Baumgartner,W. *et al.* (2007) An integrated approach to concept recognition in biomedical text. In *Proceedings of the 2nd BioCreative Challenge Evaluation Workshop*. CNIO, Madrid, pp. 257–271.
- Bunescu,R. *et al.* (2005) Comparative experiments on learning information extractors for proteins and their interactions. *Artif. Intell. Med.*, **33**, 139–155.
- Chen,L. *et al.* (2005) Gene name ambiguity of eukaryotic nomenclatures. *Bioinformatics*, **21**, 248–256.
- Hahn,U. *et al.* (2008) An overview of JCoRE, the JULIE Lab UIMA Component Repository. In *Proceedings of the LREC’08 Workshop ‘Towards Enhanced Interoperability for Large HLT Systems: UIMA for NLP’*. European Language Resources Association, pp. 1–7.
- Hakenberg,J. (2007) What’s in a gene name? Automated refinement of gene name dictionaries. In *Proceedings of the BioNLP Workshop at ACL 2007*. Association for Computational Linguistics, pp. 153–160.
- Hakenberg,J. *et al.* (2008a) Gene mention normalization and interaction extraction with context models and sentence motifs. *Genome Biol.*, **9** (Suppl. 2), S14.
- Hakenberg,J. *et al.* (2008b) Inter-species normalization of gene mentions with GNAT. *Bioinformatics*, **24**, i126–i132.
- Hanisch,D. *et al.* (2005) PROMINER: rule-based protein and gene entity recognition. *BMC Bioinform.*, **6**(Suppl. 1), S14.
- Hirschman,L. *et al.* (2005) Overview of BIOCREATIVE: critical assessment of information extraction for biology. *BMC Bioinformatics*, **6**(Suppl 1), S1.
- Hirschman,L. *et al.* (eds) (2007) In *Proceedings of the Second BioCreative Challenge Evaluation Workshop*. CNIO, Madrid. .
- Kim,J.-D. *et al.* (2003) GENIA corpus: a semantically annotated corpus for biotextmining. *Bioinformatics*, **19**, i180–i182.
- Kulick,S. *et al.* (2004) Integrated annotation for biomedical information extraction. In *Proceedings of the BioLink 2004 Workshop ‘Linking Biological Literature, Ontologies and Databases: Tools for Users’ at NAACL/HLT 2004*. Association for Computational Linguistics, pp. 61–68.
- Lafferty,J.D. *et al.* (2001) Conditional Random Fields: Probabilistic models for segmenting and labeling sequence data. In *ICML’01: Proceedings of the 18th International Conference on Machine Learning*. Omnipress, pp. 282–289.
- Leaman,R. and Gonzalez,G. (2008). BANNER: an executable survey of advances in biomedical named entity recognition. In *PSB-2008: Proceedings of the Pacific Symposium on Biocomputing 2008*. World Scientific Publishing, pp. 652–663.
- Liu,H. *et al.* (2006) BIOThESAURUS: A web-based thesaurus of protein and gene names. *Bioinformatics*, **22**, 103–105.
- Mani,I. *et al.* (2005) Protein name tagging guidelines: lessons learned. *Comp. Funct. Genomics*, **6**, 72–76.
- Morgan,A. *et al.* (2008) Overview of BIOCREATIVE II gene normalization. *Genome Biol.*, **9** (Suppl 2), S3.
- Settles,B. (2004) Biomedical named entity recognition using Conditional Random Fields and rich feature sets. In *Proceedings of the COLING 2004 NLPBA/BioNLP Workshop*. COLING, pp. 107–110.
- Tanabe,L. *et al.* (2005) GENEtag: a tagged corpus for gene/protein named entity recognition. *BMC Bioinformatics*, **6**(Suppl 1), S3.
- Tsuruoka,Y. *et al.* (2007) Learning string similarity measures for gene/protein name dictionary look-up using logistic regression. *Bioinformatics*, **23**, 2768–2774.
- Xu,H. *et al.* (2007). Gene symbol disambiguation using knowledge-based profiles. *Bioinformatics*, **23**, 1015–1022.