OXFORD UNIVERSITY PRESS | Bioinformatics

nala: text mining natural language mutations mentions

Journal:	*Bioinformatics*
Manuscript ID	BIOINF-2016-1672
Category:	Original Paper
Date Submitted by the Author:	17-Oct-2016
Complete List of Authors:	Cejuela, Juan Miguel; Technische Universität München, Department of Informatics, Bioinformatics & Computational Biology; TUM Graduate School, Center of Doctoral Studies in Informatics and its Applications (CeDoSIA) Bojchevski, Aleksandar; Technische Universität München, Department of Informatics, Bioinformatics & Computational Biology; TUM Graduate School, Center of Doctoral Studies in Informatics and its Applications (CeDoSIA) Uhlig, Carsten; Technische Universität München, Department of Informatics, Bioinformatics & Computational Biology Bekmukhametov, Rustem; Technische Universität München, Department of Informatics, Bioinformatics & Computational Biology; Microsoft Kumar Karn, Sanjeev; Technische Universität München, Department of Informatics, Bioinformatics & Computational Biology; Siemens AG Mahmuti, Shpend; Technische Universität München, Department of Informatics, Bioinformatics & Computational Biology Baghudana, Ashish; Technische Universität München, Department of Informatics, Bioinformatics & Computational Biology; Birla Institute of Technology and Science Dubey, Ankit; Technische Universität München, Department of Informatics, Bioinformatics & Computational Biology; Concur GmbH Satagopam, Venkata P.; EMBL, SCB Rost, Burkhard; Technische Universität München, Department of Informatics, Bioinformatics & Computational Biology; Institute of Advanced Study (TUM-IAS); New York Consortium on Membrane Protein Structure
Keywords:	Text mining, Natural language processing, Algorithms, Machine learning, Web services, Annotation

nala: text mining natural language mutations mentions

Juan Miguel Cejuela[1,2,*], Aleksandar Bojchevski[1,2], Carsten Uhlig[1], Rustem Bekmukhametov[1,3], Sanjeev Kumar Karn[1,4], Shpend Mahmuti[1], Ashish Baghudana[1,5], Ankit Dubey[1,6], Venkata P. Satagopam[7], & Burkhard Rost[1,8]

[1]TUM, Department of Informatics, Bioinformatics & Computational Biology - i12, Boltzmannstr. 3, 85748 Garching/Munich, Germany; [2]TUM Graduate School, Center of Doctoral Studies in Informatics and its Applications (CeDoSIA), Boltzmannstr. 11, 85748 Garching, Germany; [3]Microsoft, Advanta-A 3007 160th Ave SE, WA 98008, USA; [4]Ludwig Maximilian University, 80538 Munich & Siemens AG, Corporate Technology, 81739 Munich, Germany; [5]BITS-Pilani K. K. Birla Goa Campus, Zuarinagar, Goa, India – 403726; [6]Concur (Germany) GmbH, Lyoner Str. 15, 60528 Frankfurt am Main, Germany; [7]Luxembourg Centre for Systems Biomedicine (LCSB), University of Luxembourg, 6, avenue du Swing, L-4367 Belvaux, Luxembourg; [8]Institute of Advanced Study (TUM-IAS), Lichtenbergstr. 2a, 85748 Garching/Munich, Germany & Institute for Food and Plant Sciences WZW – Weihenstephan, Alte Akademie 8, Freising, Germany & New York Consortium on Membrane Protein Structure (NYCOMPS) & Department of Biochemistry and Molecular Biophysics, Columbia University, 701 West, 168th Street, New York, NY 10032, USA

*To whom correspondence should be addressed.

Abstract

Motivation: The extraction of sequence variants from the literature remains an important task. Existing extraction methods primarily target standard (ST) mutation mentions (e.g. “E6V”), leaving many relevant mentions in natural language (NL) largely untapped (e.g. “glutamic acid was substituted by valine at residue 6”).

Results: We introduced three new corpora suggesting named-entity recognition (NER) to be more challenging than anticipated: 28%-77% of all articles contained mentions in NL not available as ST. Our new method *nala* captured NL and ST by combining conditional random fields with word embedding features learned unsupervised from the entire PubMed. In our hands, *nala* substantially outperformed the state-of-the-art. For instance, we scanned all unique mentions in new discoveries correctly detected by any of three methods (SETH, tmVar, or *nala*). For 33% of those, only *nala* found the mention; conversely, neither SETH nor tmVar discovered anything missed by *nala*. For NL mentions the corresponding value shot up to 100% *nala*-only.

Availability: Source code, API, and corpora freely available at: http://tagtog.net/-corpora/IDP4+.

Contact: nala@rostlab.org

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Genetic variations drive biological evolution. Yet, most mutations might be harmful (Rost, 1996; Rost, et al., 2003; Sawyer, et al., 2007). Experimental studies elucidating the effects of sequence variation remain precious and expansive. Today, the important results from such studies are still published in papers. Repositories, such as OMIM (2016), rely primarily on labor-intensive and time-consuming expert curation. Searching PubMed with relevant keywords (http://1.usa.gov/1rCrKwR) brought up >1M articles; most of those (>630K) for variation in human. An equivalent search of UniProtKB/Swiss-Prot (Boutet, et al., 2016; UniProt, 2015) revealed ~13K indexed publications, and the professional version of the Human Gene Mutation Database (HGMD) (Stenson, et al., 2003)

listed ~179K mutations. These numbers sketch the immense information gap between literature and database annotations. Despite two decades of many high-level efforts to increase the incentive for authors to link their findings to databases, this gap is likely to expand even more rapidly in the future. Instead of requiring administrative overhead, text mining free literature pursues a solution that could scale and substantially narrow the gap (Krallinger, et al., 2008).

The way in which experimental results for sequence variants are reported is referred to as *mutation mentions*. Mining mutation mentions is an example of a task referred to as *named-entity recognition* (NER). We focused on the task to recognize and parse text fragments such as the following two equivalent mutation mentions: "glutamic acid was substituted by valine at residue 6" or "p.6E>V". The two differ only in their syntax: the first is written in natural language (NL), the second follows a standardized format (ST).

Existing extraction methods primarily target simple and standardized mutation mentions. MutationFinder (MF) (Caporaso, et al., 2007) uses a large set of regular expressions to recognize single nucleotide or amino acid variants (SNVs/SAVs) written in simple ST form and slightly more complex semi-standard (SST) form (e.g. "Glu 6 to Val" or "glutamic acid for valine 6"). SETH (Thomas, et al., 2014) implements a formal grammar to cover rare cases or deviations from the HGVS nomenclature (den Dunnen, et al., 2016). SETH also incorporates deprecated versions of the nomenclature to recognize other short sequence variations such as insertions and deletions (*indels*). tmVar (Wei, et al., 2013) has introduced probabilistic methods and recognizes ST mentions for a large variety of mutation types: SNPs/SAVs, insertions, deletions, frameshifts, and duplications. Existing methods are reviewed in detail elsewhere (Jimeno Yepes and Verspoor, 2014). Mapping the variant E6V to a particular sequence, e.g. that of hemoglobin S in human with the SWISS-PROT identifier *hbb_human* and relating it to sickle cell anemia (SKCA) and finally identifying that the variants is actually at position 7 in the sequence, i.e. should have been named E7V, are all essential steps toward "parsing the meaning" of the annotation. We ignored these mapping problems in this work. Instead, our work focused on presenting the first comprehensive study of the significance of natural language mutation mentions. Our new method completed the picture by recognizing different mutation types (for both genes and proteins) written in simple form or complex natural language.

2 Materials and Methods

2.1 Classification of mutation mentions: ST, SST, and NL

There is no single reliable classification of natural language (NL) or standard (ST) mutation mention. Some annotators might consider *"alanine 27 substitution for valine"* as NL because it does not follow the standard HGVS nomenclature. Others might consider it as standard or semi standard (SST) because simple regular expressions might capture this mention. Previous mutation extraction methods primarily used regular expressions and did not capture long mutation mentions.

As an operational definition, we considered any long mention that was not recognized by previous methods as NL, any mention that resembled the HGVS nomenclature as ST, and any mention in between as SST. We defined the following if-else chain algorithm to capture this idea: given a mutation mention, if it matches custom regular expressions or those from tmVar, then it is ST; else if it has 5 or more words or contains 2 or more English-dictionary words, then it is NL; else if it contains 1 English-dictionary word, then it is SST; else it is ST (examples in Table 1).

Table 1. Classification of mutation mentions

Class	Examples	*MF*	*SETH*	*tmVar*
ST	Q115P; Asp8Asn	yes	yes	yes
	c.925delA; g.3912G>C	no	yes	yes
	c.388+3insT;	no	no	yes
	delPhe1388; F33fsins; IVS3(+1)	no	no	no
SST	3992-9g-->a mutation; codon 92, TAC-->TAT	no	no	yes
	Gly 18 to Lys; leucine for arginine 90	yes	yes	no
	G643 to A	no	no	no
NL	glycine to arginine substitution at codon 20	yes	yes	no
	glycine was substituted by lysine at residue 18	no	no	no
	deletion of 10 and 8 residues from the N- and C-terminals	no	no	no

Examples of mutation mentions of increasing level of complexity as found in the literature (*ST: standard*; *SST: semi-standard*; *NL: natural language*). The columns *MF*, *SETH*, and *tmVar* indicate if the methods MutationFinder, SETH, and tmVar, respectively, recognize the examples listed.

2.2 Evaluation measures

A named entity was successfully *extracted* if its *text offsets* (character positions in a text-string) were correctly identified (tp: *true positive*). We considered two modes for *tp*: *exact* matching (two entities matched if their text offsets were *identical*) and *partial* matching (text offsets *overlapped*). Any other prediction was considered as a *false positive* (fp) and any missed entity as a *false negative* (fn). Partial matching is more suitable to evaluate NL mentions lacking well-defined boundaries. For instance, for the following mention *"[changed a highly conserved] glutamine at residue 115 to a proline"*, we did not distinguish between a solution with and without the words in brackets, because we focused on the extraction of the mention not on that of additional annotations (here "*highly conserved*"). We computed performance for all cases and for the subclasses (ST, SST, and NL). A test entity of subclass X was considered as correctly identified if any predicted entity matched. We then used the standard evaluation measures for named-entity recognition, namely, *precision* (P: $tp/tp+fp$), *recall* (R: $tp/tp+fn$), and *F-Measure* (F: $2*(P*R)/(P+R)$). Lastly, we calculated the standard error (StdErr) within a corpus and across corpora. Within a corpus, we computed the StdErr by randomly selecting 15% of the test data without replacement in 1000 (n) bootstrap samples. With $\langle x \rangle$ as the overall performance for the entire test set and x_i for subset i, we computed:

$$\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \langle x \rangle)^2} \qquad StdErr = \frac{\sigma}{\sqrt{n}} \tag{1}$$

Across corpora, we did not merge documents. Rather, we computed the mean of P, R, and F between the considered corpora, and computed the StdErr of the mean without subsampling.

2.3 Previous corpora

Some well-known corpora annotate mutation mentions and specific text offsets, including: SETH (Thomas, et al., 2014), tmVar (Wei, et al., 2013), and Variome (Verspoor, et al., 2013). All corpora contain different mutation types, including SNPs, frameshifts, or deletions. The tmVar corpus also contains rsids (reference SNP ID numbers). SETH and

tmVar are composed of abstracts, whereas Variome annotates full-text articles.

The Variome corpus annotates many vague mentions (e.g., "*de novo* mutation" or "large deletion"). We used the acronym *Variome120* to refer to the Variome subset that annotates position-specific variants; it contains 118 mentions described by the original Variome corpus (Jimeno Yepes and Verspoor, 2014) plus two new annotations with reference to both a DNA and a protein mutation.

2.4 Three new corpora: IDP4, nala, and nala_discoveries

We annotated three new corpora (*IDP4*, *nala*, and *nala_discoveries*) at different times and with slightly different objectives substantially enriching the status quo. All three were annotated with the tool *tagtog* (Cejuela, et al., 2014). The differences were as follows.

2.4.1 *IDP4* corpus

IDP4 focused on mapping mutations to sequences. We annotated the entities *Mutation*, *Organism*, and *GGP* (gene or gene product), as well as, relations between *GGP* and both *Mutation* and *Organism*. We included abstract-only and full-text documents. Documents were selected in four steps. (1) Include particular organisms/sources (*Homo sapiens, Arabidopsis thaliana, Drosophila melanogaster, Caenorhabditis elegans, Schizosaccharomyces pombe, Saccharomyces cerevisiae, Mus musculus, Rattus norvegicus*, and *HIV*). (2) Collect the PubMed identifiers linked from SWISS-PROT (Boutet, et al., 2016) when citing the keywords variation or mutagenesis. (3) Accept all abstracts that contain any of five keywords (*mutation, variation, insertion, deletion, SNP*). (4) Retrieve full-text articles through keyword *open access* (on PubMed Central).

Our method needed mutation mentions with three components: (1) W (word): a clear word or pattern giving the variant and its type (W is binary, i.e. present or not), (2) L (letter): giving the mutated nucleotides or residues (L is binary, i.e. present or not), and (3) P (position): giving the sequence location of the variation (P has three values: exact, vague, or no, i.e. not applicable). For example, P=exact as in "mutation of Tyr-838" or "Del 1473-IVS16(+2)" and P=vague as in "placed immediately downstream of I444" or "at the carboxyl end".

We distinguished two cases: (1) W=yes, L=yes, P=yes|vague, e.g. "p.Phe54Ser", "Arg-Thr insertion between 160 and 161 residues", or "(499)leucine (TTA) to isoleucine (ATA)"; (2) W=yes, L=no, P=yes, e.g. "point mutation at amino acid 444", "SNPs affecting residues, 282, 319, and 333". The rationale was that we could assign to the missing nucleotide/residue the unknown value *X*. We also annotated total gene knockouts ("Δ/Δ"), deletions of subparts ("deleted C1 domain"), or deletions of larger regions ("deletions of chromosome 9p22.3"). We considered those positions as specific. Moreover, we annotated rsids.

We measured the agreement between annotators (inter-annotator agreement: F_IAA) as proxy for the consistency of the annotations. For IDP4, we observed F_IAA=91% for all mutation mentions and F_IAA=78% for NL mentions. Four annotators participated.

2.4.2 *nala* corpus

The *nala* corpus focused on NL mutation mentions. We annotated only abstracts. We found abstracts to contain higher densities (number of mentions/number of words) of NL mentions than full articles. In particular, the IDP4, Variome, and Variome120 corpora all contained more NL mentions per word in abstracts than in their full texts (relative ratios of 5.5, 1.6, and 3.8, respectively). The document selection process was similar to that for the *IDP4* corpus. However, we applied an active learning approach to build the corpus and our new method in parallel (details below).

Annotating NL mentions strictly following our *IDP4* corpus guidelines was more challenging. For example, mutation positions were often referenced indirectly in other sentences than the variant and often in different paragraphs or otherwise were vague positions. In particular, we relaxed the rules more for insertions and deletions, e.g., "2-bp deletion in exon 6", "somatic 16-bp deletion", or "in-frame insertion of 45 nucleotides". Another unique feature of the *nala* corpus was the annotation of genetic markers such as "D17S250". For the *nala* corpus we refrained from enforcing the annotation of organisms or GGP terms. The GNormPlus tagger (Wei, et al., 2015) automatically annotated gene/protein terms.

A randomly chosen "blind" set with 90 abstracts was removed from the *nala* corpus and was exclusively used for testing, representing 15% of the entire *nala* corpus. We optimized the size of this test set such that the standard error estimate plateaued. We called this set *nala_known*.

Despite the explicit focus on NL mentions, ST mentions dominated the *nala* corpus (presumably because they are so much easier to annotate): 1097 ST mentions (52%) vs. 841 NL mentions (40%), and 170 SST mentions (8%). Thus, the *nala_known* set benchmarked both ST and NL mentions (SST mentions were underrepresented). Three experts annotated *nala*; their agreement was higher for ST than for NL mentions (F_IAA=92% for all vs. F_IAA=78% for NL mentions).

2.4.3 *nala_discoveries* corpus

We introduced another novel corpus, *nala_discoveries*, to measure the success of automatic tagging for papers that describe "new discoveries". This concept is best explained through the difference to our generic *nala* corpus: there we picked the PubMed articles beginning from identifiers of genes and proteins that had already been described experimentally and corresponded to annotations in SWISS-PROT (Boutet, et al., 2016). We had not realized how crucial this constraint was until we created a new corpus after the first version of the manuscript had been written up. In fact, the usage of previously-indexed articles and knowledge has been common practice in the field, e.g. for SNPs indexed by dbSNP or HGVS-compliant mentions (SETH corpus), disease- and mutation-specific MeSH terms indexed by PubMed (tmVar corpus), mutation-specific citations indexed by SWISS-PROT (IDP4 and nala). The Variome corpus did directly search on PubMed Central only but was restricted in scope to the three most common Lynch syndrome genes. For *nala_discoveries*, we systematically selected all articles from a particular journal and time period. We found all articles in PubMed using the keyword *mutation* and published between 2013 and 2016 in Nature (Editors, 2016), Science (Franklin, 1907), or Cell (Press, 2016) without any further filter (exact search: http://bit.ly/2aHthKP). To keep the workload in check, we randomly selected abstracts with at least one mutation mention (any form) and stopped at 60 abstracts with *at least one NL* mention. We annotated with the guidelines used for *IDP4* and *nala*. Compared to other corpora, we observed more large-scale mutations (e.g., chromosomal translocations) and significant differences in the semantics of mutation mentions. We used *nala_discoveries* exclusively for evaluation after the completion of our new method. The numbers for nala_discoveries were: 104 ST mentions (48%), 71 NL mentions (33%), and 40 SST mentions (19%). The corpus *nala_discoveries* effectively benchmarked all mention classes (incl. SST) and was annotated by the same three annotators as the nala corpus.

2.5 New method: *nala*

Conditional random fields (CRFs) (Lafferty, et al., 2001) were the basis for our new method *nala*. CRFs have been common in name-entity recognition (Settles and Burr, 2004; Wei, et al., 2013; Wei, et al., 2015). We employed the python-crfsuite implementation (tpeng, 2015), a python binding of the *CRFSuite* C++ library (Okazaki, 2007). We used standard function features such as token stems, word patterns, prefix and suffix characters, presence of numbers, or the word belonging to term dictionaries such as nucleotides, amino acids, or other biological common entities. On top, we used our in-house implementation of the *tmVar tokenizer* (Wei, et al., 2013), with the difference that we did not split tokens upon case changes at the beginning of a sentence (e.g. "The" instead of "T" + "he"). For the NL mentions, the traditional BIEO token labeling outperformed our implementation of the 11 *tmVar* labels. We also added post-processing (PstPrc) rules such as fixing small boundary problems (for example "+1858C>T" instead of "1858C>T"). We introduced additional PstPrc rules that can be switched on or off by users (e.g. annotate rsids or not). These rules significantly impacted performance (Results).

We built the *nala* corpus and method in parallel through iterative active learning that proceeded as follows. We implemented a base version of the method (*nala_1)* and trained it on the IDP4 corpus (*iteration_1* training set). For the first implementation, we reused the features from tmVar. For all following iterations (*iteration_t*), we used the previous nala model (*nala_t-1*) and a high-recall set of regular expressions to select documents that contained non-ST mentions. One of the authors ascertained that the selected documents contained at least one NL mention. In each iteration step, we arbitrarily selected ten documents. These were pre-annotated by *nala_t-1* and then posted to the tagtog annotation tool used for expert review and refinement of the pre-annotations. Finally, the reviewed annotations were saved as *iteration_t*.

After each training step, we assessed new features or tuned model parameters in 5-fold cross-validation. Often, annotators selected iteration documents that had annotation errors (missing entities, wrong offsets, or false positive predictions), precisely to learn those. We trained the final method exclusively on the *nala* corpus without using *IDP4*, due to two reasons. Firstly, NL mentions were learned considerably better when training on the *nala* corpus alone. Secondly, although ST mentions were learned slightly better with the inclusion of *IDP4*, the small improvement did not justify the complexity of training and running two separate models (ST and NL).

Word embedding features (WE) were the most important contribution to our new method. Such *WE* features had already been successfully used for biomedical named-entity recognition (Guo, et al., 2014; Passos, et al., 2014; Seok, et al., 2016; Tang, et al., 2014). Specifically, we used neural networks with the CBOW architecture (continuous bag of words) (Mikolov, et al., 2013) and trained on the entire MEDLINE/PubMed set of abstracts until mid 2015. We used a window size of 10 and a dimension D of 100. Tokens were converted to lowercase and digits were normalized to 0. For each token, the generated vector of 100 real values was translated into 100 features. The real values were used as weights in the CRF features, e.g.: word_embedding[0]=0.00492302.

2.6 Methods for comparison

We compared *nala* with two state-of-the-art methods, namely SETH and tmVar. To run SETH locally, we slightly modified the original scala code to print out the results in brat format (Stenetorp, et al., 2014). To run tmVar we used its official API (NCBI, 2015). We could not benchmark the tmVar API on the tmVar test set, because it had been trained on this set.

For each method we evaluated its default and its *best* performance. To compute the *best* performance, we removed some test annotations and predictions that are due to arbitrary annotation guidelines of the individual corpora. For example, the *best* performance of tmVar on the SETH corpus disregards rsids. In this instance, the tmVar method predicts rsids but the SETH corpus does not have those consistently annotated (9 out of the total 69). Analogously, nala predicts many actual NL mentions that are, however, not annotated in the SETH, tmVar, or Variome120 corpora. In total, we apply three filters at prediction time: rsids, genetic markers, and, for the nala method only, the use or not of WE features. WE features drive the performance and recall of NL mentions (details below) but turning them off improves the precision of nala in the ST-scoped corpora.

For all methods, the standard error between their default and *best* performance (on a same corpus) was consistently and substantially larger than the standard error within the corpus. This reflects that differences in annotation guidelines are indeed significant. Consequently, we report (Results) the averages of a default and best performance and take their standard error (individual results in Supplementary Tables S1-S5).

3 Results and Discussion

3.1 Natural language (NL) mutation mentions important.

The *Variome120* and *IDP4* corpora (unbiased towards mention forms) had significantly higher fractions of NL vs. ST or SST mentions (8% and 6%, respectively; Fig. 1, grayed out bars) than *SETH* and *tmVar* (4% and 2%, respectively). When removing mention repetitions, the fraction of unique NL mentions increased to 17% and 13% respectively (Fig. 1, highlighted bars). Exceptionally, the *Variome* corpus contained the largest fraction of SST mentions (53% and 19% with and without repetitions, respectively). NL mentions were even more predominant for abstracts (12% in *Variome120* and 13% in *IDP4* with mention repetitions and 29% and 17% without repetitions). The *nala* corpus, introduced here, was built with a higher fraction of NL mentions (40% with repetitions and 49% without repetitions). All these corpora relied on well-annotated genes and proteins (indexed articles). In contrast, the *nala_discoveries* corpus randomly sampled abstracts of publications without considering previous functional annotations (no previous indices). It contained the largest percentage of combined NL+SST mentions (52% with repetitions and 65% without repetitions).

We analyzed the annotations of three corpora (*IDP4*, *Variome*, and *Variome120*) to find out how many experimental results will methods miss that only use ST or some SST mentions. About 28%-36% of all abstracts contained at least one NL mention not existing in ST form (Table 2). The corresponding per-mention fractions were lower: 13%-27% of the mentions in abstracts existed in NL without simpler translation (Table 2). For *nala_discoveries* the numbers were substantially higher: 67%-77% (per-document) and 43%-51% (per-mention).

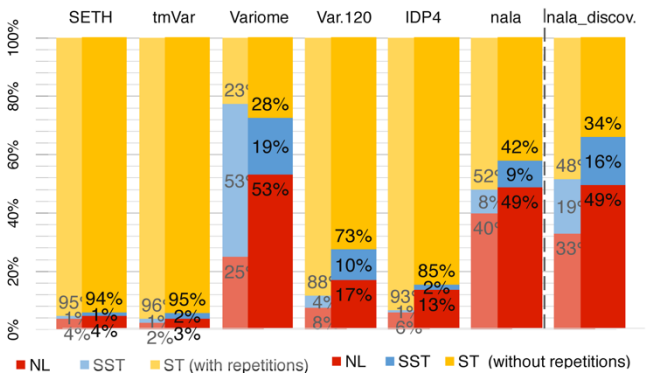

Fig. 1: Natural language (NL) mutation mentions important to consider. What type of mutation mentions dominates annotated corpora that somehow sample the literature: standard (ST, e.g. E6V), semi-standard (SST), or natural language (NL)? Grayed out bars indicate counts with repetitions, full bars unique mentions (e.g. E6V occurring twice in the same paper, is counted twice for the grayed out values and only once per paper for the others). The *Variome*, *Variome120*, *IDP4*, and *nala_discoveries* corpora assembled different representations of NL mentions. The dashed line separates corpora with papers describing well-known, well-indexed genes and proteins (left of dashed line: *SETH*, *tmVar*, *Variome*, *Variome120*, *IDP4*, and *nala_known*) and articles describing more recent discoveries that still have to be indexed in databases (right of dashed line: *nala_discoveries*).

Overall, our three new corpora (*IDP4*, *nala*, and *nala_discoveries*) accumulated the largest collection of mutation mentions: 826 documents (72 full-text), 627,953 tokens, and 5,660 mutation annotations (1,110 of those in NL). In comparison, the previous *SETH*, *tmVar*, and *Variome120* corpora combined collect: 1,140 documents (10 full-text), with 355,518 tokens, and 2,933 mutation annotations (216 NL). In other words, this work singlehandedly boosted the available resources between two and seven times.

Table 2. Significance of NL mentions

	IDP4		Variome	Var.120	nala_discoveries		
Annotator*	(1)	(2)			(1)	(2)	(3)
Documents	30%	42%	22%	33%	78%	62%	77%
Mentions	14%	19%	6%	40%	52%	39%	49%

Percentages of documents (3rd row) or mentions (4th row) that contain at least one NL (natural language) or SST (semi-standard) for which no ST (standard) mention exists in the same text. *Two different annotators were compared for the corpus *IDP4*; three different annotators were compared for the corpus *nala_discoveries*.

3.2 New method nala performed top throughout.

In our hands, our new *nala* method matched or improved over the performance of previous tools for the extraction of standard (ST) mutation mentions and significantly outperformed the status-quo in the extraction of natural language (NL) mutation mentions (Fig. 2). This baseline was very clear and simple to establish, because it was valid for all different types of evaluations we carried out. We found it more difficult to gauge what users have to expect from *nala*, and from *nala* in comparison to other methods. The reasons for the obfuscation that we could not fully eschew were in the plethora of bias for existing corpora.

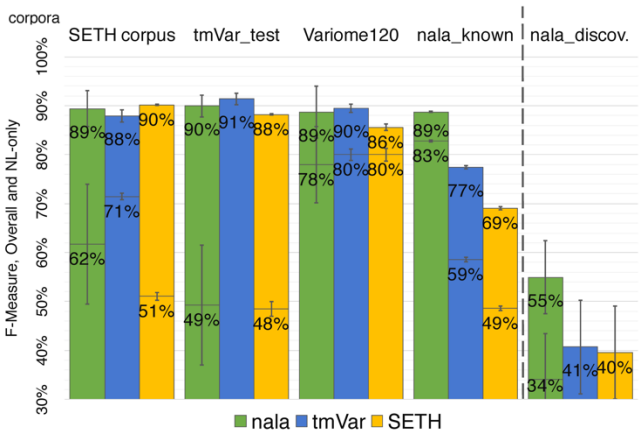

Fig. 2: *nala* performed well for all corpora. The bars give two different results: values above the horizontal lines in bars reflect the F-measures for all mentions, while values below the horizontal lines in bars reflect the F-measures for the subset of NL-mentions in the corpus (high error bars indicate corpora with few NL mentions). The exception was the result for the method tmVar on the corpus tmVar_test, which was taken from the original publication of the method in which no result was reported for NL-only (Wei, et al., 2013). That publication reports only *exact matching* performance, i.e. its *overlapping* performance might be higher than shown here. *nala* consistently matched or outperformed other top-of-the-line methods in well-indexed corpora (*SetsKnown;* left of dashed line) and substantially improved over the *status quo* in recent non-indexed discoveries (*nala_discoveries*; right of dashed line). The F-measures of tmVar and SETH for NL-only on *nala_discoveries* was essentially zero (two rightmost bars).

We tried to simplify by grouping results into those for previously indexed mutations (*SetsKnown* corpora: *SETH*, *tmVar_test*, *Variome120*, and *nala_known*; Supplementary Table S6) and those without prior knowledge (*nala_discoveries)*. To establish the performance on well-annotated genes and proteins, an average of the *SetsKnown* corpora might provide the least biased estimate: the *nala* method overall obtained F=89±3 compared to the highest performing competitor, i.e. *tmVar* with F=87±3 (Table 3). In contrast, to establish how well text mining works for randomly chosen or for all articles, we best use the *nala_discoveries* corpus: the *nala* method reached F=55±7 compared to the highest performing competitors *SETH* and *tmVar* with F=41±10 (Table 3).

Table 3. Previously indexed vs. new discoveries

method	*SetsKnowns* (indexed texts)	*nala_discoveries* (no indices)
	F ± StdErr	F ± StdErr
nala	89 ± 3	55 ± 7
tmVar	87 ± 3	41 ± 10
SETH	83 ± 5	40 ± 10

F-Measure (F) for methods *nala*, *tmVar*, and *SETH*, on corpora with previously indexed articles for mutation mentions (*SetsKnown*, corpora: *SETH*, *tmVar_test*, *Variome120*, *nala_known*) and a corpus directly sampled from PubMed without index (*nala_discoveries*).

Our new method *nala* essentially constituted a superset for the other two top methods in the following sense. The mutations correctly detected by *tmVar* and *SETH* were also found by nala. On top, *nala* correctly detected many mutations that had been missed by both other methods (Supplementary Fig. S1). Specifically, we looked at the subset of mentions

correctly detected by any of the three methods (without considering repetitions, e.g. counting the detection of E6V only once per publication): 12% (*SetsKnown* corpora) and 33% (*nala_discoveries*) of mentions were exclusively found by *nala* (Fig. 3). In contrast, only 1% and 0% (*SetsKnown* and *nala_discoveries)* were exclusively found by *tmVar*; *SETH* did not add any exclusive detection. Moreover, 50% (*SetsKnown*) and 100% (*nala_discoveries*) of natural language mentions were exclusively found by *nala* and only *tmVar* found 2% of novel NL mentions in the *SetsKnown*.

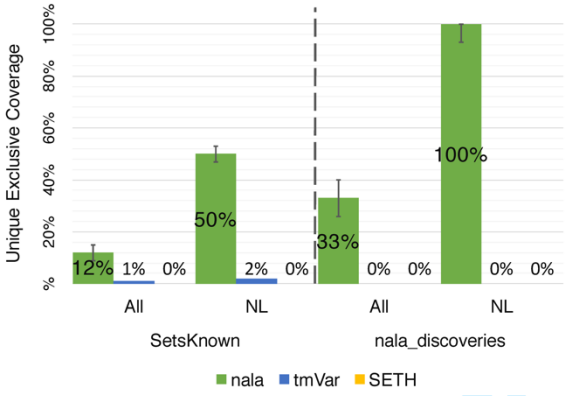

Fig. 3: *nala* could fully replace other methods. Here, we looked at the following subset of all mentions. For each publication we considered all the mentions correctly identified by one of the top three methods and kept only the findings unique in each publication. We then asked what percentage of those had been identified uniquely by one of the methods distinguishing between all mentions and NL-only mentions. For instance, for all corpora containing publications of genes and proteins indexed in the databases (SetsKnown), 1% of the mentions were detected only by *tmVar* and 12% only by *nala*, while *SETH* found no mention in this data set that *nala* had not detected. One the other end, only *nala* correctly detected NL-only mentions in papers reporting discoveries on genes/proteins not indexed in databases (100% bar on right triplet).

3.3 Word embedding (WE) features boosted performance.

The Word Embedding (WE) features contributed significantly to the success of *nala* (Fig. 4). WE features improved performance for all mention types, most importantly for NL mentions (from F(WE=off)=70 to F(WE=on)=83 on *nala_known* corpus *and from* F(WE=off)=5 to F(WE=on)=34 on *nala_discoveries corpus*). In particular, WE vastly improved recall (Supplementary Table S8). All other features by the *nala* method were specific to mutation mentions and resulted from a laborious expert optimization. In contrast, WE features constituted unsupervised data, i.e. can be adopted with minor modifications to any task or corpus.

3.4 Variants not mapped to sequence.

This work did not study the considerably more difficult problem of uniquely mapping mutation mentions to their respective biological sequences necessary for database curation. Methods recently appeared to this end (Mahmood, et al., 2016; Ravikumar, et al., 2015; Vohra and Biggin, 2013). However, all methods mapping variants to sequence still primarily target SNVs/SAVs. Next, we plan to extend the new corpora with exhaustive mapping annotations and to adapt the *nala* method to better cope with large-scale variations (predominant in *nala_discoveries*). As a practical use, we plan to research the performance of *nala* to effectively map HIV mutation mentions from whole PubMed (Davey, et al., 2014).

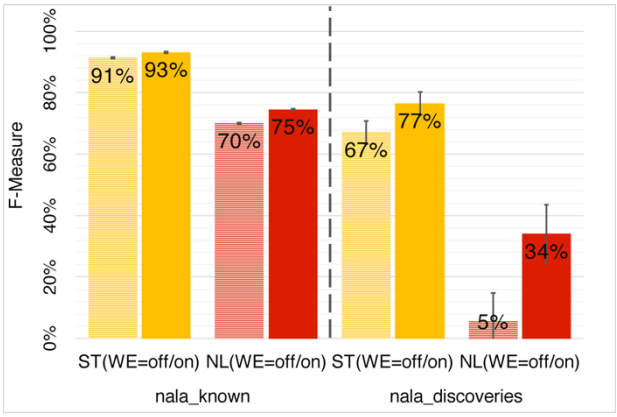

Fig. 4: Word embedding (WE) features crucial for success. The inclusion of WE features (WE=on vs. WE=off) substantially improved performance for both *nala_known* (texts previously indexed) and *nala_discoveries* (no previous indices). The increase in performance was highest for NL mentions, but for ST mentions it was also significant.

4 Conclusion

Previous accounts (Jimeno Yepes and Verspoor, 2014; Thomas, et al., 2014; Wei, et al., 2013) suggested that the strict named-entity recognition (NER) of mutation mentions constitutes a solved problem. Reported levels of performances were of F>85. Despite this optimistic perspective, the same authors (Caporaso, et al., 2007; Jimeno Yepes and Verspoor, 2014) observed that tools were unable to identify many mutations for database curation. Our work shed some light on this apparent paradox: 1) many mutation mentions use natural language (NL) and previous tools often failed to recognize those because they focused on standard (ST) forms, 2) existing corpora and methods primarily only treated articles that had been previously indexed for variations, i.e. previous knowledge existed. According to our analysis, the percentage of publications with at least one NL mention not available in ST form ranged from 28%-36% for work on genes and proteins with extensive prior work (indexed articles: *SetsKnown*) to about 67%-77% for new discoveries (no previous indices: *nala_discoveries)* that are too new to correspond to heavily indexed databases, yet (Table 2). Thus, in this view, most mentions can only be captured by methods parsing NL.

We presented a new method *nala* designed to handle NL and ST mentions. In particular, word embedding (WE) features boosted performance for NL mentions (Fig. 4). In our hands, *nala* at least matched the best existing tools for *corpora, SetsKnown*, dominated by ST mentions (F(nala)=89±3 vs. F(tmVar)=87±3, Table 3). For a corpus, *nala_discoveries*, randomly sampled directly from PubMed without filtering on previous indices, in which NL and SST (semi-standard) mentions dominated, *nala* was substantially better than existing methods (F(nala)=55±7 vs. F(SETH, tmVar)=40-41±10, Table 3). What to expect as a final user, 89 or 55? The answer depends on what you know about the article you are looking for. For *older* articles, point mutations, or *indels*, the current performance of all methods may be sufficient. For novel work or large-scale mutations, *nala* can find many mutation mentions that are missed by other methods (Fig. 3). However, we still miss about half of all variants described in the literature.

An important contribution of this work was the addition of three new corpora (*IDP4*, *nala_known*, and *nala_discoveries*). These three new corpora accumulated the largest collection of mutation mentions: 826 documents (72 full texts) and 5660 mutation annotations (1110 NL). For

several aspects of the NER task this resource increased what exists manifold. We released the new method as an open source python library and as API service and made the new corpora freely available: http://tagtog.net/-corpora/IDP4+

Acknowledgements

Thanks to Tim Karl for invaluable help with hardware and software; to Inga Weise for excellent administrative support; to Tatyana Goldberg for assistance in preparing and submitting the manuscript; to Maria Biryukov and Esteban Peguero Sánchez for helpful comments on the manuscript.

Funding

This work was supported by a grant from the Alexander von Humboldt foundation through the German Federal Ministry for Education and Research (BMBF).

Conflict of Interest: none declared.

References

2016. Online Mendelian Inheritance in Man, OMIM®. http://omim.org/. (2016/5/13 date last accessed)|.

Boutet, E., *et al.* UniProtKB/Swiss-Prot, the Manually Annotated Section of the UniProt KnowledgeBase: How to Use the Entry View. *Methods Mol Biol* 2016;1374:23-54.

Caporaso, J.G., *et al.* MutationFinder: a high-performance system for extracting point mutation mentions from text. *Bioinformatics* 2007;23(14):1862-1865.

Caporaso, J.G., *et al.* INTRINSIC EVALUATION OF TEXT MINING TOOLS MAY NOT PREDICT PERFORMANCE ON REALISTIC TASKS. In, *Biocomputing 2008*. 2007.

Cejuela, J.M., *et al.* tagtog: interactive and text-mining-assisted annotation of gene mentions in PLOS full-text articles. *Database (Oxford)* 2014;2014(0):bau033.

Davey, N.E., *et al.* The HIV mutation browser: a resource for human immunodeficiency virus mutagenesis and polymorphism data. *PLoS Comput Biol* 2014;10(12):e1003951.

den Dunnen, J.T., *et al.* HGVS Recommendations for the Description of Sequence Variants: 2016 Update. *Hum. Mutat.* 2016;37(6):564-569.

Editors, N. 2016. About Nature. http://www.nature.com/nature/about/index.html

Franklin, C.L. Magazine Science. *Science* 1907;25(645):746.

Guo, J., *et al.* Revisiting Embedding Features for Simple Semi-supervised Learning. In, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2014.

Jimeno Yepes, A. and Verspoor, K. Mutation extraction tools can be combined for robust recognition of genetic variants in the literature. *F1000Res.* 2014;3:18.

Krallinger, M., Valencia, A. and Hirschman, L. Linking genes to literature: text mining, information extraction, and retrieval applications for biology. *Genome Biol* 2008;9 Suppl 2:S8.

Lafferty, J.D., McCallum, A. and Pereira, F.C.N. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In, *Proceedings of the Eighteenth International Conference on Machine Learning*. Morgan Kaufmann Publishers Inc.; 2001. p. 282-289.

Mahmood, A.S.M.A., *et al.* DiMeX: A Text Mining System for Mutation-Disease Association Extraction. *PLoS One* 2016;11(4):e0152725.

Mikolov, T., *et al.* Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* 2013.

NCBI. 2015. NCBI Text Mining Tools. http://www.ncbi.nlm.nih.gov/CBBresearch/Lu/Demo/tmTools/. (2016/5/13 date last accessed)|.

Okazaki, N. 2007. CRFsuite - A fast implementation of Conditional Random Fields (CRFs). http://www.chokkan.org/software/crfsuite/. (2016/5/13 date last accessed)|.

Passos, A., *et al.* Lexicon Infused Phrase Embeddings for Named Entity Resolution. In, *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*. 2014.

Press, C. 2016. Cell Contact. http://www.cell.com/contact

Ravikumar, K.E., *et al.* Text mining facilitates database curation - extraction of mutation-disease associations from Bio-medical literature. *BMC Bioinformatics* 2015;16(1).

Rost, B. PHD: predicting one-dimensional protein structure by profile-based neural networks. *Methods Enzymol* 1996;266:525-539.

Rost, B., *et al.* Automatic prediction of protein function. *Cell Mol Life Sci* 2003;60(12):2637-2650.

Sawyer, S.A., *et al.* Prevalence of positive selection among nearly neutral amino acid replacements in Drosophila. *Proceedings of the National Academy of Sciences* 2007;104(16):6504-6510.

Seok, M., *et al.* Named Entity Recognition using Word Embedding as a Feature. *International Journal of Software Engineering and Its Applications* 2016;10(2):93-104.

Settles, B. and Burr, S. Biomedical named entity recognition using conditional random fields and rich feature sets. In, *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications - JNLPBA '04*. 2004.

Stenetorp, P., Pyysalo, S. and Topić, G. 2014. Standoff format - brat rapid annotation tool. http://brat.nlplab.org/standoff.html. (2016/5/13 date last accessed)|.

Stenson, P.D., *et al.* Human Gene Mutation Database (HGMD ®): 2003 update. *Hum. Mutat.* 2003;21(6):577-581.

Tang, B., *et al.* Evaluating word representation features in biomedical named entity recognition tasks. *Biomed Res. Int.* 2014;2014:240403.

Thomas, P., *et al.* 2014. SETH - SNP Extraction Tool for Human Variations. https://rockt.github.io/SETH/. (2016/5/13 date last accessed)|.

tpeng. 2015. tpeng/python-crfsuite. https://github.com/tpeng/python-crfsuite. (2016/5/13 date last accessed)|.

UniProt, C. UniProt: a hub for protein information. *Nucleic Acids Res.* 2015;43(Database issue):D204-212.

Verspoor, K., *et al.* Annotating the biomedical literature for the human variome. *Database* 2013;2013:bat019.

Vohra, S. and Biggin, P.C. Mutationmapper: a tool to aid the mapping of protein mutation data. *PLoS One* 2013;8(8):e71711.

Wei, C.-H., *et al.* tmVar: a text mining approach for extracting sequence variants in biomedical literature. *Bioinformatics* 2013;29(11):1433-1439.

Wei, C.-H., Kao, H.-Y. and Lu, Z. GNormPlus: An Integrative Approach for Tagging Genes, Gene Families, and Protein Domains. *Biomed Res. Int.* 2015;2015:918710.