Automatic Identification and Normalisation of Physical Measurements in Scientific Literature

Luca Foppiano
FOPPIANO.Luca@nims.go.jp
National Institute for Materials Science (NIMS)
Tsukuba, Japan

Laurent Romary
laurent.romary@inria.fr
Inria
Paris, France

Masashi Ishii
ISHII.Masashi@nims.go.jp
National Institute for Materials Science (NIMS)
Tsukuba, Japan

Mikiko Tanifuji
TANIFUJI.Mikiko@nims.go.jp
National Institute for Materials Science (NIMS)
Tsukuba, Japan

ABSTRACT

We present Grobid-quantities, an open source application for extracting and normalising measurements from scientific and patent literature. Tools of this kind, aiming to understand and make unstructured information accessible, represent the building blocks for large-scale Text and Data Mining (TDM) systems. Grobid-quantities is a module built on top of Grobid [5], a machine learning framework for parsing and structuring PDF documents. Designed to process large quantities of data, it provides a robust implementation accessible in batch mode or via a REST API. The machine learning engine architecture follows the cascade approach, where each model is specialised in the resolution of a specific task. The models are trained using CRF (Conditional Random Field) algorithm [11] for extracting quantities (atomic values, intervals and lists), units (such as length, weight) and different value representations (numeric, alphabetic or scientific notation). Identified measurements are normalised according to the International System of Units (SI). Thanks to its stable recall and reliable precision, Grobid-quantities has been integrated as the measurement-extraction engine in various TDM projects, such as Marve (Measurement Context Extraction from Text), for extracting semantic measurements and meaning in Earth Science [9]. At the National Institute for Materials Science in Japan (NIMS), it is used in an ongoing project to discover new superconducting materials. Normalised materials characteristics (such as critical temperature, pressure) extracted from scientific literature are a key resource for materials informatics (MI) [8].

CCS CONCEPTS

• **Applied computing** → **Document analysis**; *Document metadata*; *Format and notation.*

DocEng'19, September 23–26, 2019, Berlin, DE

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/1122445.1122456

KEYWORDS

Machine Learning, TDM, Measurements, Physical quantities, Units of measurements

ACM Reference Format:
Luca Foppiano, Laurent Romary, Masashi Ishii, and Mikiko Tanifuji. 2019. Automatic Identification and Normalisation of Physical Measurements in Scientific Literature. In *Proceedings of DocEng'19: ACM Symposium on Document Engineering (DocEng'19).* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

The data overflow in scientific publications makes rapid access to relevant information a challenging issue, for both researchers and readers. One of the essential element found in scientific literature is the physical quantity or measurement, which combine quantification of units (such as grams or micrometres) and quantified object or substances. The automatic extraction of measurements has been studied for many years. Nowadays, although the technology has been evolved, there are still several challenges to overcome: (1) natural language and writing style have varieties of expressions (for example length can be expressed as m, meter, metre). (2) Overlaps between the different units of measurement (*pico Henry* inductance and acidity share the same notation, *pH*). (3) The physical quantities or measurements are scalable by accompanying units (e.g., 1 pl. = 453.6 g), meaning that value and unit combination and its normalisation are necessary for semantic recognition. The need for a precise automatic generation of databases from physical measurements is common to a wide range of domains.

In this paper, we present Grobid-quantities, an Open Source application for identifying, parsing and normalising measurements from scientific and patent literature. Using Conditional Random Field (CRF) [11], it provides a machine learning framework for extracting information in a robust manner, and then normalise them toward the International System of Units (SI). This article is organised as follow. In Section 2 we introduce related work. Then, we describe the system in Section 3 and report its evaluation results in Sections 4. Use cases and future scopes are described in Section 5. Section 6 concludes this paper.

2 RELATED WORK

Attempts to extract measurements from text have been made using rule-based (formal grammars engines, look-ups in terminological

databases) and ML approaches. A known commercial tool, Quantalyze[1], was reported by [9] showing weak recall and supporting only a limited subset of units [3]. Another approach [1], using GATE (General Architecture for Text Engineering), addressed the identification of numeric properties from patents. [2] investigated issues applied to Russian-derived languages. These approaches lack either the generalisation to an extensive corpus or deal mainly with specific languages. [4] described an attempt to recognise units by looking up terms from an ontology, using ML in combination with pattern matching and string metrics. Other ML-based approaches exist, although limited to specific domains: [10] and [7] describe measurements extraction from experimental results in biology and nanocrystal device development, respectively. Our work is not restricted to a specific domain or subset of measurements and includes a normalisation process.

3 SYSTEM DESCRIPTION

Grobid-quantities is a Java application, based on Grobid (GeneRation Of BIbliographic Data) [5], a machine learning framework for parsing and structuring raw documents such as PDF or plain text. Grobid-quantities is designed for large-scale processing tasks in batch or via a web REST API. Results information are standardised and can be stored in databases or visualised on PDF, via the obtained GROBID build-in positional coordinates.

3.1 Data model

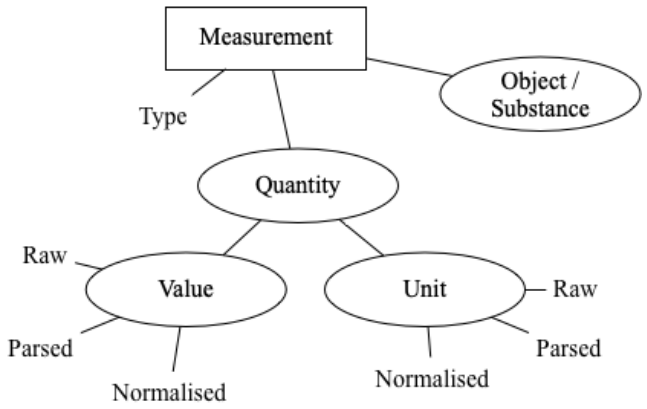

Figure 1: Schema of the data model.

The data model (Figure 1) lay its foundation on the concept of *Measurement*, which links an object or a substance with one or more *quantities*. We defined four *Measurements* types: (a) atomic, in case of a single measurement (e.g., 10 grams). (b) interval (*from 3 to 5 km*) and (c) range (100 ± 4 mm) for continuous values, and, (d) a list of discrete values. A *Quantity* links the quantitative value and the unit. At this stage we do not support probability distribution of ranges and intervals. Since data extracted from PDFs unavoidably present irregular tokens from wrong UTF-8 encoding or missing fonts, we designed this model to allow partial results. The *Value* and *Unit* entities allow three different representations (Figure 1): *Raw* as appear in input, *Parsed* unifies the value into the numerical expression, and the unit with its properties (system, type). Finally, *Normalised* contains the transformed unit and values to the SI system. *Value* object supports four types of representations: numeric (2, 1000), alphabetic (two, thousand), scientific notation ($3 \cdot 10^5$), and time, which is also expression of measurements. Units objects are organised following the SI, which allows representing units as products of simpler compounds (e.g. m/s to $m \cdot s^{-1}$) further decomposed as triples (prefix, base and power).

[1] https://www.quantalyze.com/

3.2 Architecture

The system takes in input text or PDF and performs three steps: (a) tokenisation, (b) measurement extraction and parsing and (c) quantity normalisation. The details of each step are summarised as follows.

3.2.1 Tokenisation. This process splits input data into tokens. Grobid-quantities uses a two-phase tokenisation: (1) first it splits by punctuation marks, then (2) each resulting token is re-tokenised to separate adjacent digits and alphanumeric characters. Given the example `25m^2`, first returns a list `[25m, ^, 2]` and then recursively divides `25m` as `[25, m]` resulting in `[25, m, ^, 2]`.

3.2.2 Extraction. The tokenised data is labelled by three models, applied in cascade: the *Quantities* CRF model determines appropriate unit and value tags. Results are processed in cascade by the respective *Units* and *Values* CRF models as illustrated in Figure 2.

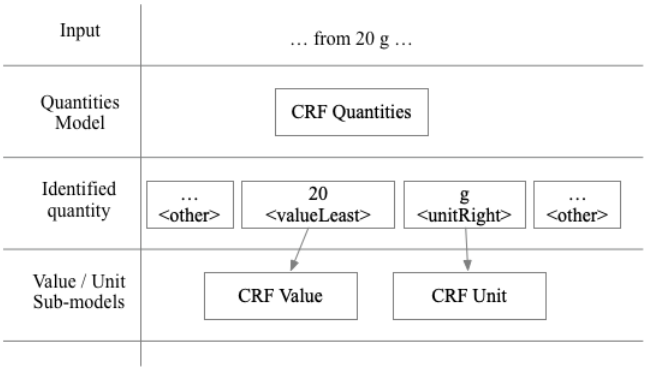

Figure 2: The cascade approach in applied CRF models. The Quantities model recognises value and units which are passed, respectively, to Values and Units CRF sub-models for further extraction.

As illustrated in Table 1, *Quantities* CRF model uses additional labels (such as <unitLeft>, <unitRight> for units) to correctly reconstruct complex objects from the flat structure obtained from the sequence labelling output.

Previous work (Section 2) presented extensive use of databases or ontologies. In our solution, we used a similar approach. We created a list of units (in English, French and German) with their characteristics: system (SI base, SI derived, imperial, ...) and type (volume, length, ...), and their representations: notations (m^3, `m^3`), lemmas (cubic meter, cubic metre) and inflections (cubic meters, cubic metres). We made this list available through the *Unit Lexicon*,

Table 1: Labels description for the CRF model for Quantities. In bold the token the label refers to.

Label	Description	Example
<valueAtomic>	value of an atomic quantity	**2** m
<valueLeast>	least value in an interval	from **2** m
<valueMost>	max value in an interval	up to **7** m
<valueBase>	base value in a range	**20** ± 7 m
<valueRange>	range value in a range	20 ± **7** m
<valueList>	list of quantities	**2**, **3** and **10** m
<unitLeft>	left-attached unit	**pH** 2
<unitRight>	right-attached unit	2 **m**
<other>	everything else	-

which offers unit lookups by properties (such as notation, lemma, inflexion). A second gazetteer was created to allow the transformation of alphabetic values in numeric ones (for example, twenty-one to 21).

Features in the *Quantities* CRF model are generated using standard information (preceding and following tokens, presence of capital, digits). Orthogonal features are obtained through the *Unit Lexicon*, like a *Boolean* indicating whether a token is a known unit or not. Typographical information (such as format, fonts, subscript/superscript) are ignored.

The *Units* CRF model works at character level and uses the *Unit Lexicon* to highlight known units or prefixes. The input tokens are parsed and transformed to a product of triples (prefix, base, power) as shown in Table 2. For example `Kg/mm`^{2}, corresponds to $Kg \cdot mm^{-2}$ and becomes `[(K, g, 1), (m, m, -2)]` as product of triples.

Table 2: Labels description for the CRF *Units* model. The example shows in bold the part referred by the label.

Label	Description	Example
<prefix>	prefix of the unit	**k**m^2
<base>	unit base	k**m**^{2}
<pow>	unit power	km$^{\mathbf{2}}$
<other>	everything else	-

We then use the structured triples to fetch the corresponding (system, type) information from the *Unit Lexicon* and attach them to the resulting object. At the moment we do not exploit any contextual information related to the paper domain to resolve ambiguous units. In parallel, the CRF *Values* model unifies the format of identified values into numerical formats. It supports four types: numeric, alphabetic, scientific notation, and time expression (see Table 3). Each type is treated with different parsers, namely alphabetic expressions are looked up in the word-to-number gazetteer, scientific notations are parsed and calculated mathematically. Time expressions are processed using the built-in Date Grobid model [5].

3.2.3 Normalisation. The measurements extracted are transformed to the base SI unit (grams to kg, Celsius to Kelvin, and so on). We used an external Java library called Units of Measurement which provides a set of standard interfaces and implementations for safely handling units and quantities. Manipulating measurements with transformations often lead to common mistakes due to wrong rounding and approximations. At the time this paper is being written, the final revised version of this library has been accepted under the Java Standardisation Process JSR-385.

Table 3: Labels description for the CRF model for Values. In bold an example of tokens for the specific label recognise.

Label	Description	Example
<number>	numeric value / coefficient	$\mathbf{2.5} \cdot 10^5$
<alpha>	alphabetic value	**twenty**
<time>	time expression	in **1970-01-02**
<base>	base in scientific notation	$2.5 \cdot \mathbf{10}^5$
<pow>	exponent in scientific notation	$2.5 \cdot 10^{\mathbf{5}}$
<other>	everything else	-

4 EVALUATION AND RESULTS

We trained and evaluated our system using a corpus of 32 Open Access (OA) English articles retrieved from different domains such as medicine, robotics, astronomy, and physiology. The corpus contains additionally three patents translated in English, French and German. Three people annotated the corpus, and each document was cross-checked. The corpus, although small, is public, documented and open to external contributions. We partitioned training and evaluation data using 80/20 proportion. We estimated precision, recall and F1-score for each model, using the evaluation framework built-in in Grobid. These measure indices are calculated at three different levels: token-level, field-level and instance-level. Given a fragment with their predicted and expected labels. While token-level scores are calculated independently for each token, field-level scores are calculated for each continuous sequence of tokens under the same label (a sequence of several tokens which all belong to the same labelled chunk, e.g. a unit), finally instance-level is the aggregated score of fields in the whole paragraph [8].

Table 4: Evaluation scores for *Quantities* CRF model (precision, recall and F1-score).

Label	Token-level			Field-level		
	P	R	F1	P	R	F1
<unitLeft>	98.94	95.23	97.05	97.8	95.11	96.43
<unitRight>	66.67	66.67	66.67	59.09	54.17	56.52
<valueAtomic>	86.63	87.81	87.22	87.39	87.17	87.28
<valueBase>	95.12	100	97.5	94.12	94.12	94.12
<valueLeast>	82.81	65.43	73.1	81.89	67.1	73.76
<valueList>	77.69	56.63	65.51	76.06	58.06	65.85
<valueMost>	78.05	73.44	75.68	81.68	64.46	72.05
<valueRange>	96.67	100	98.31	94.44	94.44	94.44
average	88.81	85.08	**86.9**	89.59	84.6	**87.02**

As shown in Tables 4 we obtained average F1-score of 86.9% and 87.02% for token and field level respectively. The low score for *<valueLists>* and *<unitRight>* suggests that more examples of that

kind are required. The highest F1-score for *<valueRange>* above 90% is reasonable, because of limited variety in expressions for a Range as compared with normal Intervals. The recall at instance level is 68.49%, indicating that more than half of the evaluated paragraphs were correctly labelled.

Table 5: Units CRF model evaluation results (precision, recall and F1-score).

Label	Token-level			Field-level		
	P	R	F1	P	R	F1
<base>	97.52	92.49	94.94	77	82.8	79.79
<pow>	81.82	90	85.71	73.33	84.62	78.57
<prefix>	62.79	93.1	75	70.27	92.86	80
average	90.64	92.37	**91.49**	75	85.07	**79.72**

Table 5 shows that *Units* CRF models F1-score is 91.49% and 79.72% for token and field level, respectively. The F1-score dropped by 10% from token to field-level. We noticed a strong bias toward *base-only* unit examples. While this make sense, because simple units are statistically more frequent, it indicates that more examples covering complex units are needed. Instance-level recall is not reported for *Units* and *Values* because it overlap with field-level as most of the instances are composed by one field.

Table 6: Values CRF model evaluation results (precision, recall and F1-score).

Label	Token-level			Field-level		
	P	R	F1	P	R	F1
<alpha>	100	100	100	100	100	100
<base>	86.67	46.43	60.47	66.67	42.86	52.17
<number>	91.75	97.45	94.51	90.43	97.2	93.69
<pow>	92.86	65	76.47	77.78	50	60.87
<time>	89.83	86.89	88.33	54.55	75	63.16
average	91.85	90.95	**91.4**	86.18	86.75	**86.47**

Finally, Table 6 indicate that *Values* CRF model has average f1-score of 91.4% and 86.47% for token and field level, respectively. We noticed that both *<base>*, *<pow>* and *<time>* have lower f1-score, suggesting that more contextual information should be introduced as features, like tokens preceding or following the value.

5 APPLICATIONS

Recently, the normalised data extraction is strongly required in materials research, because an inverse problem in which high-performance materials are predicted from properties is expected to be solved with well-organised big data. At the National Institute for Materials Science (NIMS), a project to discover new superconducting materials from scientific literature is in progress. The system being developed relies on Grobid-quantities to extract and normalise superconducting properties, such as critical temperature (Tc) with units of mK and K and critical pressure expressed with units of Pa, MPa, and GPa [8]. Grobid-quantities was showcased also in a TDM prototype (within the scope of the French national-wide ISTEX [6] project) where it provided measurement annotations used to prototype a quantities-based semantic search[2]. Finally, another use was made in a system for extracting semantic measurements and meaning in Earth Science, Marve [9].

6 CONCLUSION

In this paper, we presented Grobid-quantities, a system for extracting and normalising measurement from scientific and patent literature. Results are promising, and the integration in real applications proved a consolidated level of maturity. There is still the need to have more training data, in particular for the *Quantities* and *Units* CRF models, respectively. In the future, we plan to improve the extraction by introducing embeddings and recurrent neural networks, like Bi-LSTM+CRF as a replacement for the statistical CRF models. In the same scope we plan to add more contextualised information (article domain) and additional layout features (like for example superscript/subscripts). The project, the training data and the documentation are accessible on Github at the address http://github.com/kermitt2/grobid-quantities.

ACKNOWLEDGMENTS

Our warmest thanks to Patrice Lopez (author of Grobid [5] and other Open Source TDM tools), who initiated and supported Grobid-quantities. Thanks our colleagues at NIMS Thaer M. Dieb, and Akira Suzuki for the support received. Finally, thanks to Units of Measurements's contributors[3].

REFERENCES

[1] Milan Agatonovic, Niraj Aswani, Kalina Bontcheva, Hamish Cunningham, Thomas Heitz, Yaoyong Li, Ian Roberts, and Valentin Tablan. 2008. Large-scale, parallel automatic patent annotation. In *Proceedings of the 1st ACM workshop on Patent information retrieval*. ACM, 1–8.

[2] Skopinava AM and Lobanov BM. 2013. Processing of quantitative exPressions with units of measurement in scientific texts as aPPlied to Belarusian and russian text-to-sPeech synthesis. (2013).

[3] Hidir Aras, René Hackl-Sommer, Michael Schwantner, and Mustafa Sofean. 2014. Applications and Challenges of Text Mining with Patents.. In *IPaMin@ KONVENS*.

[4] Soumia Lilia Berrahou, Patrice Buche, Juliette Dibie-Barthélemy, and Mathieu Roche. [n. d.]. How to Extract Unit of Measure in Scientific Documents?.

[5] Contributors 2008. GROBID (GeneRation Of BIbliographic Data). https://github.com/kermitt2/grobid. arXiv:1:dir:6a298c1b2008913d62e01e5bc967510500f80710

[6] André Dazy. 2014. ISTEX: a powerful project for scientific and technical electronic resources archives. *Insights* 27, 3 (2014).

[7] Thaer M Dieb, Masaharu Yoshioka, Shinjiro Hara, and Marcus C Newton. 2015. Framework for automatic information extraction from research papers on nanocrystal devices. *Beilstein journal of nanotechnology* 6, 1 (2015), 1872–1882.

[8] Luca Foppiano, M. Dieb Thaer, Akira Suzuki, and Masashi Ishii. 2019. Proposal for Automatic Extraction Framework of Superconductors Related Information from Scientific Literature. In *Letters and Technology News, vol. 119, no. 66, SC2019-1 (no.66)*, Vol. 119. Tsukuba, 1–5. ISSN: 2432-6380.

[9] Kyle Hundman and Chris A Mattmann. 2017. Measurement Context Extraction from Text: Discovering Opportunities and Gaps in Earth Science. *arXiv preprint arXiv:1710.04312* (2017).

[10] Yanna Shen Kang and Mehmet Kayaalp. 2013. Extracting laboratory test information from biomedical text. *Journal of pathology informatics* 4 (Aug. 2013), 23–23. https://doi.org/10.4103/2153-3539.117450

[11] John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. (2001).

[2]The demo can be accessed at https://traces1.inria.fr/istex_sample/

[3]https://github.com/orgs/unitsofmeasurement/people