# Automatic Identification and Normalisation of Physical Measurements in Scientific Literature

Luca Foppiano, Laurent Romary, Masashi Ishii, Mikiko Tanifuji

# 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 [6] [13], 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 [12] 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 [10]. 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) [9].

## CCS CONCEPTS

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


*DocEng '19, September 23–26, 2019, Berlin, Germany*

ACM ISBN 978-1-4503-6887-2/19/09…$15.00
https://doi.org/10.1145/3342558.3345411

## 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 *ACM Symposium on Document Engineering 2019 (DocEng '19), September 23–26, 2019, Berlin, Germany*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3342558.3345411

## 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) [12], 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, lookups in terminological

databases) and ML approaches. A known commercial tool, Quantalyze[1], was reported by [10] 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: [11] and [8] 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 web application, based on Grobid (GeneRation Of BIbliographic Data) [6] [13], a machine learning framework for parsing and structuring raw documents such as PDF or plain text. Grobid-quantities is designed to process large quantity of data, via web, through a REST (Representation State Transfer) API or locally, via the file-system (batch mode). Output information are standardised as stand-off annotations, and they can be stored in databases or indexed in search engines. Each annotation can be visualised on top of PDFs using the GROBID build-in positional coordinates.

### 3.1 Data model

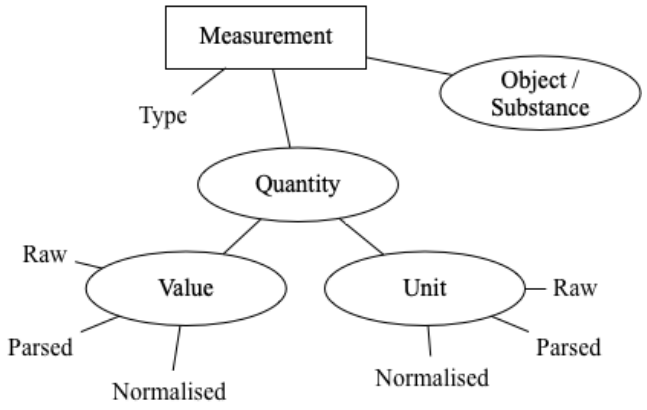


**Figure 1: Schema of the data model.**

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 ($100 \pm 4$ mm) for continuous values, and, (d) a list of discrete values. A *Quantity* links the quantitative value and the unit. 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 an expression of measurement. 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).

### 3.2 Architecture

The system takes in input text or PDF (the content is extracted in a structured way using the Grobid framework) 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 tokens are passed through three ML models, in cascade: first the *Quantities* CRF model determines appropriate unit and value tags. Results are further processed by the respective *Units* and *Values* CRF models as illustrated in Figure 2.

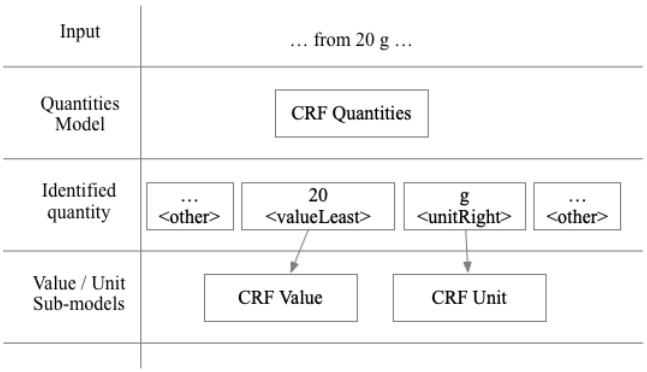


**Figure 2: The cascade approach in applied CRF models. The Quantities model recognises value and units which are passed, respectively, to Values and Units CRF sub-models for further extraction.**

Table 1 describes the labels predicted by the *Quantities* CRF model. Notice that, to reconstruct complex structured objects from the flat sequence generated by the engine, additional labels are necessary (such as <unitLeft>, <unitRight>, for units).

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

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

**Table 1: Labels description for the Quantities CRF model. In bold are highlighted specific examples.**

| 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 | - |

(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*, 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 from 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 and 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 Units CRF model. In bold are highlighted specific examples.**

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

We then use the structured triples to fetch the corresponding information (system, type) from the *Unit Lexicon* and attach them to the resulting object. This implementation processes the unit characters using right-to-left order. Priority modifiers, such as parenthesis, are ignored. They are generally not frequent in units expressions, and require a more complex logic.

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). Different techniques are applied for each type: alphabetic expressions are looked up in the word-to-number gazetteer, scientific notations are parsed and calculated mathematically. Time expressions are further segmented using the Grobid built-in Date CRF model.

**Table 3: Labels description for the Values CRF model. In bold are highlighted specific examples.**

| 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 | - |

*3.2.3 Normalisation.* The measurements extracted are transformed to the base SI unit (grams to kg, Celsius to Kelvin, etc.). We used an external Java library called Units of Measurement [5], 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.

## 4 EVALUATION AND RESULTS

We trained and evaluated our system's models using a dataset based on 32 scientific publication (English, Open Access (OA)) and three patents (with translation in English, French and German) randomly selected from different domains such as medicine, robotics, astronomy, and physiology. The training data was generated automatically and then corrected and cross-checked by three annotators. We used 10-fold cross-validation to evaluate each CRF model, independently, and produce precision, recall and f1 scores, as summarised in Table 4.

The Quantities CRF model reported an f1 macro average of 80.14% with precision and recall of 84.93% and 76.86%, respectively. The paragraph accuracy was 68.97%, indicating that more than half of the evaluated paragraphs were correctly labelled. These scores are promising, considering the complexity of the task and the rather small size of the training corpus. In particular, `<list>` and `<unitRight>` require more example.

The Units CRF model f1 macro average was 98.86%, with precision and recall reaching 98.75% and 98.97%, respectively. Compared with our other models, performances were extremely high (more than 10% for f1 score). Such difference can be attributed to the data distribution and the lower variability of unit expressions. We analysed the training data, and we noticed that the distribution is biased toward simple units (composed by a single triple). Intuitively, this makes sense, because simple units are statistically more frequent; on the other hand, it highlights the necessity of having more complex examples in our dataset. Secondly, unit expressions appear, by nature, with lower variability, leading to the generation of more duplicates than in other models' training datasets. For example, the expressions 1% and 2% have two different values (1, 2) and the same unit (%), which would appear twice. Since we cannot alter the statistical distribution of the dataset, we would obtain better and more precise measurements of the model generalisation capabilities by using a separate and independent evaluation corpus.

**Table 4: Summary of the evaluation scores (precision, recall, F1-score) and label contribution (support) for *Quantities, Units* and *Values* CRF models, respectively.**

| Label | Precision | Recall | F1 | Support |
|---|---|---|---|---|
| **Quantities CRF model** | | | | |
| <unitLeft> | 96.76 | 94.71 | 95.71 | 2805 |
| <unitRight> | 93.06 | 72.1 | 80.02 | 120 |
| <valueAtomic> | 85 | 84.77 | 84.84 | 3599 |
| <valueBase> | 78.82 | 76.52 | 77.53 | 94 |
| <valueLeast> | 85.05 | 77.39 | 80.94 | 862 |
| <valueList> | 72.09 | 54.87 | 61.33 | 494 |
| <valueMost> | 84.09 | 73.03 | 78.07 | 878 |
| <valueRange> | 84.56 | 81.5 | 82.68 | 93 |
| all (macro avg.) | 84.93 | 76.86 | **80.14** | |
| **Unit CRF model** | | | | |
| <base> | 99.02 | 99.22 | 99.12 | 3075 |
| <pow> | 98.04 | 98.9 | 98.46 | 322 |
| <prefix> | 99.19 | 98.8 | 98.99 | 821 |
| all (macro avg.) | 98.75 | 98.97 | **98.86** | |
| **Values CRF model** | | | | |
| <alpha> | 96.64 | 98.65 | 97.62 | 826 |
| <base> | 83.06 | 69.23 | 72.77 | 58 |
| <number> | 98.01 | 99.02 | 98.52 | 3858 |
| <pow> | 76.45 | 74.67 | 74.58 | 56 |
| <time> | 72.54 | 87.83 | 79.34 | 322 |
| all (macro avg.) | 85.34 | 85.88 | **84.57** | - |

The Value CRF model scored average macro f1 at 85.64% with precision and recall at 81.82% and 93.29%, respectively. We noticed that both *<base>*, *<pow>* and *<time>* have lower f1-score. While *<base>* and *<pow>* require more training data, *<time>* expressions may overlap with *<number>* suggesting more contextual information should be introduced.

## 5 APPLICATIONS

Recently, the normalised data extraction is strongly required in materials research. The 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 [9].

Grobid-quantities was showcased in a Text and Data Mining (TDM) platform (within the scope of the French national-wide ISTEX [7] 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 [10].

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

## 6 CONCLUSION

In this paper, we presented Grobid-quantities, a system for extracting and normalising measurement from scientific and patent literature. The project, the training data and the documentation are accessible on Github[3].

Results are promising, and the integration in real production platforms proved this application reached a certain level of maturity. Our dataset, although it requires more training examples, is released as open access and can be improved from external contributors. Moreover, as previously discussed, the introduction of an end to end evaluation corpus could provide more objective evaluation results.

In the future, we plan to introduce recurrent neural networks (RNN) and embeddings for sequence labelling. In particular, contextualised embeddings, trained with values and units could improve the model generalisation. Finally, we plan to integrate domain information and additional layout features (such as superscript/subscripts) to improve unit discrimination.

## ACKNOWLEDGMENTS

Our warmest thanks to Patrice Lopez (author of Grobid and many 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[4].

## 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 [n. d.]. Units of Measurement. https://github.com/unitsofmeasurement.

[6] Contributors 2008 — 2019. GROBID (GeneRation Of BIbliographic Data). https://github.com/kermitt2/grobid. swh:1:dir:6a298c1b2008913d62e01e5bc967510500f80710.

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

[8] 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.

[9] 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.

[10] Kyle Hundman and Chris A Mattmann. 2017. Measurement Context Extraction from Text: Discovering Opportunities and Gaps in Earth Science. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM.

[11] 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

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

[13] Patrice Lopez. 2009. GROBID: Combining automatic bibliographic data recognition and term extraction for scholarship publications. In *International conference on theory and practice of digital libraries*. Springer, 473–474.

[3]http://github.com/kermitt2/grobid-quantities

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