Text Mining Methods Applied to Mathematical Texts

Yannis Haralambous

Text Mining Methods Applied to Mathematical Texts

Yannis Haralambous[1]

DECIDE - Lab-STICC - Télécom Bretagne

Abstract. Up to now, flexiform mathematical text has mainly been processed with the intention of formalizing mathematical knowledge so that proof engines can be applied to it. This approach can be compared with the symbolic approach to natural language processing, where methods of logic and knowledge representation are used to analyze linguistic phenomena. In the last two decades, a new approach to natural language processing has emerged, based on statistical methods and, in particular, data mining. This method, called text mining, aims to process large text corpora, in order to detect tendencies, to extract information, to classify documents, etc. In this paper we present *math mining*, namely the potential applications of text mining to mathematical texts. After reviewing some existing works heading in that direction, we formulate and describe several roadmap suggestions for the use and applications of statistical methods to mathematical text processing: (1) using terms instead of words as the basic unit of text processing, (2) using topics instead of subjects ("topics" in the sense of "topic models" in natural language processing, and "subjects" in the sense of various mathematical subject classifications), (3) using and correlating various graphs extracted from mathematical corpora, (4) use paraphrastic redundancy, etc. The purpose of this talk is to give a glimpse on potential applications of the math mining approach on large mathematical corpora, such as `arXiv.org`.

1 Mathematics and natural language

The goal of "Data mining" (or "Knowledge discovery in databases") is to extract potentially useful and previously unknown knowledge from huge amounts of data. According to [1], [10] makes a distinction between *text mining*, which is the process of discovering and extracting knowledge from *unstructured* data, and data mining, which discovers knowledge from structured data. Under this view, text mining comprises three major activities: information retrieval, to gather relevant texts; information extraction, to identify and extract a range of specific types of information from texts of interest; and the usual mining algorithms, to find associations among the pieces of information extracted from many different texts.

But what is "unstructured data"? It is data with more-or-less hidden structure that has to be extracted, to be useful. Also there are so many different ways of extracting structure that it can be merely considered as an interpretation among several others, and the way your interprete your data depends on the

final application you have in mind. A typical example of unstructured data is text in *natural language*.

And how about mathematical text? Consider the following six statements:

1. "There is no triplet of positive nonzero integers for which the sum of the cubes of the first two is equal to the cube of the third."
2. "The equation $a^3 + b^3 = c^3$ has no solution in the set of positive nonzero integers."
3. "$(a, b, c) \in A \subset \mathbb{N}^3, a^3 + b^3 = c^3 \Rightarrow A = \{(0, 0, 0)\}$."
4. "$\forall p \, \forall q \, \forall r \, (\mathrm{sum}(\mathrm{cube}(p), \mathrm{cube}(q)) = \mathrm{cube}(r)) \wedge \mathrm{inN}(p) \wedge \mathrm{inN}(q) \wedge \mathrm{inN}(r) \models (p = 0 \wedge q = 0 \wedge r = 0)$."
5. "Fermat's Last Theorem is true for $n = 3$."
6. "Le dernier théorème de Fermat est vrai en degré 3."

They carry more-or-less the same knowledge and natural language is used at various degrees and in different ways: in the first case the statement is given in natural language only, the second is a mixture of a formula and text (actually a textual statement on the solutions of the equation in the formula), the third is written in standard mathematical notation (assuming knowledge of notation $\mathbb{N}$ for the set natural numbers), the fourth is a representation of statement in FOL (assuming predicates "sum," "cube," and "inN," the latter meaning "belongs to $\mathbb{N}$"). The last two refer to a known result, namely Fermat's last theorem: note that in English the capitalization of words "Last Theorem" shows the fact that they constitute a named entity.

1.1 Various degrees of language formality

[15] defines the word *flexiform* as an adjective to describe the fact that a representation is of flexible formality, i.e., can contain both *informal* (i.e., appealing to a human reader), and *formal* (i.e., supporting syntax-driven reasoning processes) means. According to [17], there are many steps between "informal" and "formal," Informality does not necessarily contradict rigorous style, and symbolic notation is not necessarily formal." and also "Rigorous natural language, often called "mathematical vernacular," has the potential to be understood by a machine."

Besides flexiform and "rigorous" language there is also "controlled" (a language between formal and natural, with specific constraints, like the simplified versions of English used for technical documentation, which can be automatically translated to other language without information loss, cd. 1.2) and "specialized" (= using the jargon of a given scientific or technical discipline).

But what about mathematical text?

Note that in a sentence like "A is abelian," the symbol A acts as a noun and has the syntactic role of a NP. It is denoting a mathematical object (probably a group), given earlier in the text. We deduce that (flexiform) mathematical text can be analyzed by traditional NLP methods: morphology, syntax, semantics, pragmatics. There have been two important works in this area: [2] and [27]:

1. [2] uses the HPSG (head-driven phrase structure grammar) approach for describing syntax and λ-DRT (λ-discourse representation theory) for semantics.
2. [27] mentions a "parser module," for POS tagging and syntactic analysis, and then also uses DRT for semantics.

Both [2] and [27] aim to rigorously analyze mathematical text in order to use theorem provers subsequently. This corresponds to the "symbolic" approach to NLP. Unfortunately they provide only limited results:

- [2]: "Insgesamt analysiert der implementierte Prototyp zwar nur einen kleinen Ausschnitt des von uns betrachteten Textes — das 2. Kapitel von [Bartle and Sherbert, 1982] — vollständig ... Die im Detail untersuchten drei Theoreme (und Beweise) zeigen viele repräsentative Probleme für die Verarbeitung mathematischer Texte auf."
- [27]: "Given the enormous complexity of the entire problem, much implementation work is to be done to enable Vip to read and understand, say all the proofs of Hardy & Wright's textbook on elementary number theory ... At the time of writing we are only aware of Vip being able to completely process two example constructions."

Let us recall the differences between symbolic and statistical NLP methods (from [19]):

- "*Symbolic approaches* to NLP perform deep analysis of linguistic phenomena and are based on explicit representation of facts about language through well-understood knowledge representation schemes and associated algorithms."
  "... A good example of symbolic approaches is seen in logic- or rule-based systems. In logic-based systems, the symbolic structure is usually in the form of logic propositions. Manipulations of such structures are defined by inference procedures that are generally truth preserving. Rule-based systems usually consist of a set of rules, an inference engine, and a workspace or working memory. Knowledge is represented as facts or rules in the rule-base. The inference engine repeatedly selects a rule whose condition is satisfied and executes the rule."
- "*Statistical approaches* employ various mathematical techniques and often use large text corpora to develop approximate generalized models of linguistic phenomena based on actual examples of these phenomena provided by the text corpora without adding significant linguistic or world knowledge. In contrast to symbolic approaches, statistical approaches use observable data as the primary source of evidence."

At this point we may ask how to deal with mathematical text. Shall we aim for an "approximative generalized model" of mathematical text? Are we going to "approximate Bourbaki"? Isn't that heresy?

Let us (re)view possible strategies.

1.2 Possible strategies for flexiform mathematical text

- Strategy #1 (for the brave): Use a controlled language from the very beginning.
- Strategy #2: Use XML markup to structure as much as possible.
- Strategy #3: Use a visual language to structure as much as possible.
- Strategy #4: Use statistical methods.

Strategy #1: Use a controlled language According to Wikipedia, "*Controlled natural languages* (CNLs) are subsets of natural languages, obtained by restricting the grammar and vocabulary in order to reduce or eliminate ambiguity and complexity. ... [Some of them] have a formal logical basis, i.e., they have a formal syntax and semantics, and can be mapped to an existing formal language, such as first-order logic. Thus, those languages can be used as knowledge-representation languages, and writing of those languages is supported by fully automatic consistency and redundancy checks, query answering, etc."

This is, for example, the case of Mizar [24] and the *Journal of Formalized Mathematics*, which is a special, esthetically beautiful way of writing mathematics, even though it is not (yet) the way to write a paper or a thesis, for most of us. Citing [9], "Most users of mathematics are not versed in formal mathematics and, even if they were, it is not yet clear that it could support their activities adequately."

Strategy #2: use XML markup This is the OMDOC approach [15]. A mathematical document is an XML file. The mathematical formulas are represented in OpenMath (which is content-based, as opposed to formal systems that are semantic-based and allow proof-checking), but there is also provision for formal proofs in the system. LaTeX to OMDOC translation can be done (almost) automatically. It is an ongoing project for the arXiv corpus [7]. An IDE for OMDOC, Sentido [8], allows visualization of formula structure.

Strategy #3: use a visual language This is the MathLang [12, 13] approach: in TeXmacs, the user places boxes around parts of his text and formulas. Syntactical roles of text blocks or of formula parts are treated in the same way: $\boxed{\boxed{a} + \boxed{0}}$ equals $\boxed{a}$ is equivalent to $\boxed{\boxed{a} + \boxed{0} = \boxed{a}}$. The author of the text has to manually annotate (= put boxes around) his text and formulas.

Strategy #4: use statistical methods Many NLP methods exist to process ordinary text.

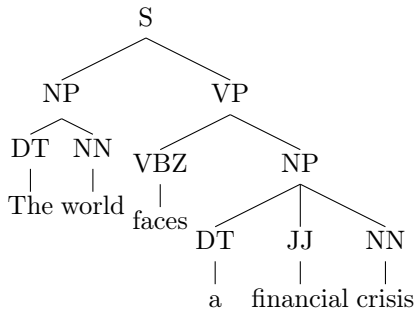
A morphological analyzer (or "POS-tagger") will parse, for example, the sentence *The world faces a financial crisis* as

The/DT
world/NN

faces/VBZ
a/DT
financial/JJ
crisis/NN

but, doing this, will also detect, that "faces" can be the plural of noun "face" or the 2nd person of singular of present of verb "to face," with given probabilities.

A syntax parser will produce the most likely tree



and hence disambiguate morphological analysis (if other trees are possible, they will be give as well, with given probabilities).

A semantic annotator will match the meaning of words in semantic resources, for example in WordNet, with given probabilities...

The noun face has 13 senses (first 8 from tagged texts)

1. (193) **face**, human face -- (the front of the human head from the forehead to the chin and ear to ear; "he washed his face"; "I wish I had seen the look on his face when he got the news")
2. (23) expression, look, aspect, facial expression, **face** -- (the feelings expressed on a person's face; "a sad expression"; "a look of triumph"; "an angry face")
3. (22) **face** -- (the general outward appearance of something; "the face of the city is changing")
4. (4) **face** -- (the striking or working surface of an implement)
5. (2) **face** -- (a part of a person that is used to refer to a person; "he looked out at a roomful of faces"; "when he returned to work he met many new faces")
6. (1) side, **face** -- (a surface forming part of the outside of an object; "he examined all sides of the crystal"; "dew dripped from the face of the leaf")
7. (1) **face** -- (the part of an animal corresponding to the human face)
8. (1) **face** -- (the side upon which the use of a thing depends (usually the most prominent surface of an object); "he dealt the cards face down")
9. grimace, **face** -- (a contorted facial expression; "she made a grimace at the prospect")
10. font, fount, typeface, **face**, case -- (a specific size and style of type within a type family)
11. **face** -- (status in the eyes of others; "he lost face")
12. boldness, nerve, brass, **face**, cheek -- (impudent aggressiveness; "I couldn't believe her boldness"; "he had the effrontery to question my honesty")
13. **face** -- (a vertical surface of a building or cliff)

The verb face has 9 senses (first 6 from tagged texts)

1. (56) confront, face up, **face** -- (deal with (something unpleasant) head on; "You must confront your problems"; "He faced the terrible consequences of his mistakes")
2. (33) confront, **face** -- (oppose, as in hostility or a competition; "You must confront your opponent"; "Jackson faced Smith in the boxing ring"; "The two enemies finally confronted each other")
3. (14) front, look, **face** -- (be oriented in a certain direction, often with respect to another reference point; be opposite to; "The house looks north"; "My backyard look onto the pond"; "The building faces the park")
4. (4) **face** -- (be opposite; "the facing page"; "the two sofas face each other")
5. (4) **face** -- (turn so as to face; turn the face in a certain direction; "Turn and face your partner now")
6. (3) confront, **face**, present -- (present somebody with something, usually to accuse or criticize; "We confronted him with the evidence"; "He was faced with all the evidence and could no longer deny his actions"; "An enormous dilemma faces us")
7. **face** -- (turn so as to expose the face; "face a playing card")
8. **face** -- (line the edge (of a garment) with a different material; "face the lapels of the jacket")
9. **face** -- (cover the front or surface of; "The building was faced with beautiful stones")

There are also several other levels of processing: pragmatics, requiring knowledge of the world; discourse processing; anaphora resolution, etc.

Many mathematical structures has been used to modelize these phenomena (starting from finite state automata and probabilistic models and going all the way to fractals, quantum mechanics and high energy physics...).

To handle flexiform mathematical text, various strategies could be used:

1. Adapt existing NLP tools.
2. Adapt corpora, i.e., convert formulas (or geometric constructions, etc.) into natural language text. For formulas, this can be done, for example, by

- converting to LaTeX,
- and then using ASTER [23] or some similar system.

3. Use a hybrid approach (adapt both corpus and tools).

Here is an example of existing work in that direction:

[25] uses frequencies of mathematical symbols as features for document classification. This is the bag-of-words approach, restricted to mathematical symbols. Why not combine features such as symbols, words or even, as we will see, *terms*? Advantages of this approach: no knowledge resource needed, symbols are easy to detect and count. Disadvantages: semantic ambiguity (which could be easily avoided by looking into surrounding terms), working on document level only.

In the remaining part of this paper we will give suggestions about how to process mathematical text using NLP methods.

2 Suggestion 1: Use terms instead of words

First of all, let us ask the fundamental question: "What is a term?"

In OMDOC [15], a (technical) term is "*a phrase representing a concept* for which a declaration exists in a content dictionary." Terms in OMDOC are tagged by the user, *termhood* property is binary.

For [6] also, "terms are linguistic representations of concepts," but, nevrtheless, one can define a termhood property which is a positive scalar. Compare for example:

mobile phone / mobile / red mobile phone / red telephone
automorphism / canonical automorphism / trivial automorphism

"Mobile phone" is the original name of a wireless telephone for outside use. This is a complex term. For brevity, the term "mobile" has progressively become an alternative version of "mobile phone," in other words: the "phone" part has become redundant. When we add the adjective "red" we do not create a new term since red color is simply a variable property of mobile phones, and red color does not change the technical specifications of the device. Nevertheless "red telephone" is a term when we mean the specific telephone line between Washington and Moscow.

In the same way, a "canonical automorphism" is a special kind of automorphism and hence is clearly a term, but "trivial automorphism" can be a specific automorphism (for example, the identity map) or simply an automorphism whose description is very simple. Therefore we can state that the termhood property of "trivial automorphism" is less than the one of "canonical automorphism."

The phenomenon when combining two terms produces a term with a completely different meaning is called a "red herring" (which is neither red, nor a herring). There are very few red herrings in mathematics, an example: child drawings, which are not (only) drawings, and certainly not drawn by children.

We see that termhood depends on the term and on the context. There are several ways of calculating it, here is how.

2.1 C-Value

For [6], a term is a string of words of the following form:

```
((Adj|Noun)+ | ((Adj|Noun)*(NounPrep)?)(Adj|Noun)*)Noun
```

A term can be nested in longer terms, take for example "graded algebra" which is contained in "differential graded algebra," contained in "commutative differential graded algebra," etc.

The *C-value* for termhood is defined as follows:

$$\text{C-value}(a) = \begin{cases} \log_2(|a|+1) \cdot f(a) & \text{if } a \text{ not nested,} \\ \log_2(|a|+1) \cdot \left(f(a) - \frac{1}{\#(T_a)} \sum_{b \in T_a} f(b) \right) & \text{otherwise,} \end{cases}$$

where f is frequency in the corpus, $|.|$ length (in words), T_a the set of terms containing a.

2.2 NC-Value

As defined above, C-value depends strictly on frequencies in the corpus. Another characteristic of terms is that they are compatible with specific modifiers and not with others. For example, in math, multi-word units starting with "easy" have lower chances of being terms, those starting with "simplicial" have strong chances of being terms.

The *NC-value* is defined as follows in [6]:

First, the weight of a word w (adj, noun or verb, preceding or following a term) is defined as

$$\text{Weight}(w) = \frac{t(w)}{n}$$

where $t(w)$ is the number of terms w appears with, and n the number of terms considered.

Then

$$\text{NC-value}(a) = 0.8\,\text{C-value}(a) + 0.2 \sum_{b \in C_a} f_a(b) \cdot \text{Weight}(b)$$

where C_a is the set of context words of a and $f_a(b)$ the frequency of b as context word of a.

2.3 Term variation

While representing the same concept, terms can have several *variants*. Term variation can be:

- orthographic: "pull-back" vs. "pullback," "neighbor" vs. "neighbour,"
- morphological: singular/plural, "Riemann space" vs. "Riemannian space,"
- lexical: "Abelian" vs. "commutative," "epimorphism" vs. "surjective morphism,"
- structural: for example, possessive usage of nouns using prepositions, like in "isomorphism of groups" vs. "group isomorphism,"

- abbreviational: "log" vs. "logarithm" (even outside formulas),
- compositional: "Wahscheinlichkeitstheorie" vs. "Theorie der Wahrscheinlichkeit,"
- acronymic: "GCD", "CW-complex" vs. "closure-finite weak-topology complex,"
- naming: "Banach space" vs. "complete normed vector space,"
- reductive: "complex conjugate function" vs. "complex conjugate" (substantification), etc.

Some of these need extra resources, others can be detected by simple transformation rules. [21] handles term variation as follows:

1. acronym acquisition
2. inflectional normalization
3. structural normalization
4. orthographic normalization

and then uses the usual method for (N)C-value calculation, after having merged variations into a standard form (called *canonical representative*).

2.4 Term and symbol interaction

In math, very often terms are used to describe symbols, or, inversely, symbols are used to denote objects represented by terms. [26] uses word-sense disambiguation methods to map symbols (in fact, simple expressions) to terms. Instead of recognizing terms, they are taken from a pre-existing taxonomy. "Simple expressions" are atomic identifiers with optional superscripts and/or subscripts. As we see, context can help both for calculating termhood as for finding relations between terms and symbols denoting them.

Word-sense disambiguation methods as in [26] can also profit from the fact that there are (domain-dependent) notational conventions so that we can calculate probabilities of term/symbol matches and inject them into the learning algorithm.

Notational conventions depend on domain but also on language: "Sei K un Körper," "Let F be a field," "Soit C un corps," ... Also, some notations become domain-independent ($\mathbb{R}, \mathbb{C}$, ...), others not ($\mathbb{P}$ can denote "projective" or "probability").

This raises the question: *Should we use placeholders for symbols?* If yes, we loose knowledge on notational conventions. If no, we have less chances of identifying formulas. We suggest using a hybrid approach. One can build the distribution of denotations given a symbol, and then attach it to placeholders.

2.5 Suggestions about dealing with context

Up to now, the context was simply a set of words neighboring a term, or a scientific domain. But terms (and symbols) belong to *documents*. And documents are *events* in the *real world*: as every event they have

– a cause (the authors),
– an intention (to spread knowledge, to become rich and famous),
– a timeline (they are created, modified),
– a (physical or virtual) support (a journal), etc.

Knowledge of 1, 3 and 4 can contribute to disambiguate a term or a symbol.

Authors are objects belonging to various graphs: co-authorship, citations, social networks... Timeline is also important since terms have an organic life span: they are born, they grow by getting used, they interact, sometimes even procreate, and sometimes die. Finally, journals sometimes alter notations or terminology.

Therefore we suggest not to consider a mathematical text as an abstract piece of pure wisdom, living in some Platonic world. After all, mathematical texts are written by people. People have their own styles, and tend to reuse the same notations and terminologies. People communicate with other people. They share ideas and knowledge. As ideas and knowledge are communicated by notation and terminology, these are (often) shared as well.

When extracting terms and finding term↔symbol matchings, a to-do list would be:

– find out who wrote the text,
– find and analyze his/her other writings,
– find his/her "friends," co-authors, the people he/she cites,
– find and analyze their writings (especially those cited)
– check the timeline of all documents considered.

3 Suggestion 2: Use topics instead of subjects

3.1 Keywords, subjects

Most math papers contain a list of keywords and a subject classification, in some scheme (for ex. AMS MSC 2010, see also [17]). For MSC, AMS suggests using one primary classification and several secondary ones. Keywords are freely chosen by the author. They express the author's opinion about the important concepts in his/her paper. The advantages of this approach are: the author is best suited to know the important keywords and subjects of his/her paper. And the disadvantages: they are global despite the fact that different parts of the paper may require specific metadata; there is subjectivity involved in their choice (difference between what our paper is about and what we would like it to be about).

3.2 Topics

In the text mining world, a *topic* is a statistical model for discovering themes occurring in a document collection. Not only can we have several topics per document, or per document section, but we can also study their interrelations,

and construct topic hierarchies or graphs. Topic models are very popular because they apply not only to the document paradigm, but also to form recognition in images, bioinformatics and other fields.

Were are some statistical models using topics:

Mixture of unigrams The simplest topic model is the *mixture of unigrams* [3]. For each document (among M) choose a topic z using some distribution $p(z)$, and then generate N words w independently from the conditional multinomial $p(w \mid z)$. The joint probability of document $\mathbf{w}$ and topic z is

$$p(\mathbf{w}, z) = p(z) \prod_{n=1}^{N} p(w_n \mid z).$$

In this model we use only one topic per document.

Latent Dirichlet Allocation A more elaborate model is *Latent Dirichlet Allocation* [3]. For each document (among M) choose a Dirichlet distribution of K topics (θ_*) based on parameters (α_*). Then for $n = 1, \ldots, N$ choose:

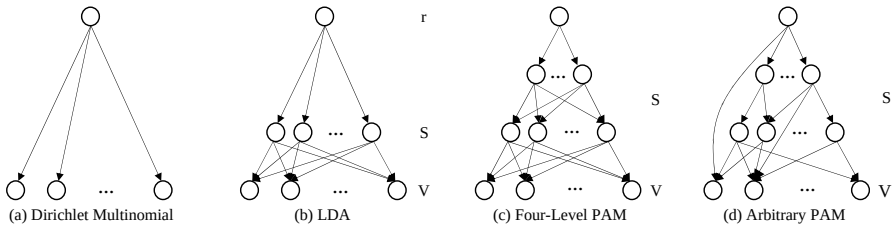
- a topic z_n from a multinomial distribution with parameters (θ_i),
- a word w_n from $p(w_n \mid z_n, \beta)$, a multinomial distribution conditioned on z_n, with parameters (β_*).

The joint probability of document $\mathbf{w}$ (N words), set of N topics $\mathbf{z}$ and topic mixture θ is

$$p(\mathbf{w}, \mathbf{z}, \theta \mid \alpha, \beta) = p(\theta \mid \alpha) \prod_{n=1}^{N} p(z_n \mid \theta) p(w_n \mid z_n, \beta).$$

In this model we have many topics per word and per document, but there is no correlation between topics.

Pachinko Allocation Model The *Pachinko Allocation Model* [18] is an enhanced version of LDA. A directed acyclic graph is constructed, whose edges are topics at various levels and words. Vertices represent dependance.



(a) Dirichlet Multinomial (b) LDA (c) Four-Level PAM (d) Arbitrary PAM

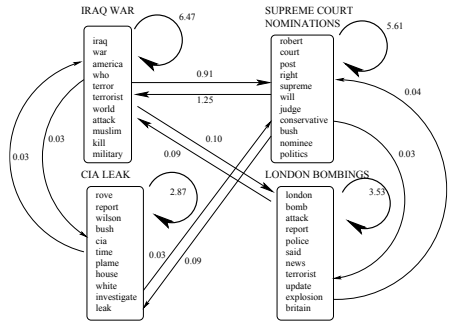
The Pachinko model is generated as follows: Suppose we have S topics. For each document, sample $\theta_{t_1}, \ldots \theta_{t_s}$ from Dirichlet distributions of parameters $\alpha_1, \ldots, \alpha_s$. Each θ_{t_i} is a multinomial distribution of topic t_i overs its children on the graph. For each word w in the document:

– sample a topic path $\mathbf{z}_w$ of length L_w. Each $z_{w,i}$ is a child of $z_{w,i-1}$ and is sampled according to the multinomial $\theta_{z_{w,i-1}}$,
– sample a word w from $\theta_{z_{w,L_w}}$.

The joint probability of a document d, the topic assignments $\mathbf{z}$ and the multinomials θ is

$$p(d, \mathbf{z}, \theta \mid \alpha) = \prod_{i=1}^{s} p(\theta_{t_i} \mid \alpha_i) \prod_{w} \left(\prod_{i=2}^{L_w} p(z_{w,i} \mid \theta_{z_{w,i-1}}) p(w \mid \theta_{z_{w,L_w}}) \right).$$

Joint topic models for text and citations [20] introduce a model (called Link-PLSA-LDA) that combines text topics and citations. The idea is that citations can contribute in capturing topicality of the document.



In this figure, citation probabilities have to be multiplied by 0.0015. Intra-topic citation probability is high.

Syntactic topic models [4] introduce a model that discovers topics that are both semantically and syntactically coherent. This model uses LDA for the semantic part and FTIC (Finite tree with independent children [5]) for the syntactic part. For the Syntactic Topic Model, the observed data are documents, each of which is a collection of dependency parse trees.

3.3 Suggestions

Using topic models, one could

– classify math documents by topics, calculated upon terms.
– calculate topic also on the section level, and combine global and local results.
– use syntactic topic model to capture dependencies between symbols and terms denoting them, etc.
– use joint topic model for text and citation, to include information on cited documents (particularly important in mathematics).
– investigate correlation between metadata introduced by authors and topics obtained.

- use Pachinko topic model and investigate correlation between mathematical domains (in an Algebraic Topology paper, will we find topics "algebra" and "topology," and how will they be interrelated?).
- create the hierarchical (dynamic) graph of mathematical topics.

And here are some potential applications:

- One can consider a paper as a vector in the space of topics. In this space we can calculate semantic/topical similarity.
- An author would be the sum of papers he/she wrote (potentially weighted by "importance"). Finding people working in the same domain would be projecting their vectors upon topics of the domain and calculating their cosine.
- Social networks and research teams are graphs of people. They become graphs in the space of topics. One could investigate correlations between graphical measures and topical ones.
- Journals and conferences can be considered as the sums of vectors of papers they publish. One could investigate their evolution over time, the correlation between their papers, etc.

3.4 "Gehirnsturm und Drang," open questions

Is there a way to infer quality of a paper using statistical methods?

And how about readability?

Can we build a service which will recommend papers for reading, based not only on topical and temporal proximity, but also on potential fertilization?

Could we keep track of everything you read and write (incl. drafts, emails, chats, etc.), of everything your friends/colleagues read and write, etc., to increase the quality of this recommendation?

Mathematics are used in other disciplines. Topics could allow us to detect parts of mathematics that have been applied elsewhere as well as those that haven't yet. Surveying them could give ideas for future work, PhDs...

Can topics complement the Mathematician's global intuition on the field?

4 Suggestion 3: Extract graphs from corpora

4.1 The structure of mathematical corpora

Some blocks of mathematical text are mainly *intradocumental*: proofs, lemmas, corollaries, acknowledgments... and others are of *interdocumental* nature: conjectures, definitions, theorems. They can be new statements, in which case they may obtain the names of their authors. Or they may be taken from other documents, in which case, besides their "names" they should contain citations. Or they may simply be mentioned more-or-less explicitly...

One can build a digraph of these statements, oriented by temporality or citation direction.

The graph of interdocumental statements is only one of the many graphs one can build out of mathematical corpora:

- graph of co-authors, institution sharing (use also the *Mathematics Genealogy Project*)
- graph of citation sharing
- graph of topic/subject/keyword sharing
- graph of denotation sharing (same symbol for same meaning)
- graph of acknowledgment sharing, etc.
- mathematical ontologies...

All of them can be naturally oriented (either by time or by intrinsic properties), and derived graphs can be built: authors, institutions, topics, symbols, funded projects, etc.

There is a research area called *graph mining*. It allows us to define measures and calculate distances between graphs. One can find vertices or subgraphs with specific good properties (for example: communities in social networks), one can also find frequent substructures. One can learn graph grammars using machine learning methods.

Under this view, a document is not an isolated node anymore, but the center of a neighborhood in the various graphs we mentioned. And the same goes for authors, institutions, topics, etc.

Not to mention that even if mathematics are eternal, mathematical corpora/communities evolve, and this evolution can be represented by successive versions of graphs.

4.2 Typical interaction between graphs and natural language

Geometric constructions (cf. [22]) can be modelled as knowledge base graphs:

- points, lines, etc. are instances of concepts in a geometry ontology,
- relations between them are instances (for example RDF triplets) of relations in that ontology.

These graphs can be compared, patterns found, etc. They can be converted into natural language (with a lot of redundancy); Redundancy can be avoided by considering inference relations. Both natural language and graphs can be used for querying.

5 Suggestion 4: Use paraphrastic redundancy

On Fig. **??** one can see the same Chinese sentence translated in many different ways into English. Statistical NLP methods can used the various translations of the same sentence to improve accuracy of knowledge extraction. Traditionally a sentence "having the same meaning" as a give one, is called a paraphrase, this is why we call information included in various translations of the same sentence, "paraphrastic redundancy." How can this be applied to mathematics?

Stating a theorem is like translating a mathematical fact into flexiform text. There are dozens of books on the same topic (Algebra, Analysis, Topology, etc.). We could improve knowledge on specific theorems or parts of theories, by comparing the various versions of the same statement in several documents, using the corresponding NLP methods.

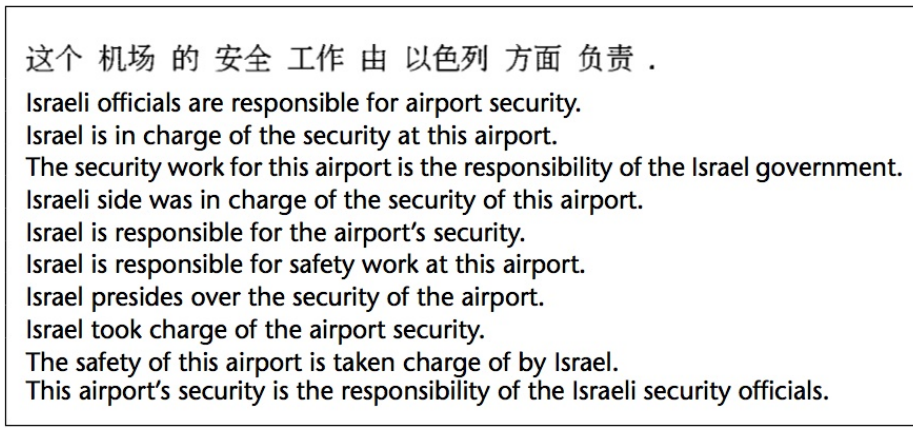


Fig. 1. Taken from [11].

6 Conclusion

While waiting for rigorous systems to analyze in depth hundreds of thousands of mathematical texts, a lot can be done by processing mathematical text using natural language tools. The results will always be "statistical," ambiguous, approximative, but "approximative" does not mean "unreliable" nor "useless." For some applications, like ("fuzzy") searching, classifying, recommending, surveying, detecting tendencies or plagiarism, etc., this may be perfectly sufficient. Statistical Mathematical Language Processing, or *Math Mining*, may be a **useful research area for the near future**.

References

1. S. Ananiadou & J. McNaught, *Text Mining for Biology and Biomedecine*, Artech House, 2006.
2. J. Baur, *Syntax und Semantik mathematischer Texte*, Diplomarbeit, Saarbrücken, 1999.
3. D. Blei, A. Y. Ng & M. I. Jordan, Latent Dirichlet Allocation, *Journal of Machine Learning Research*, 3 (2003) 993–1022.
4. J. Boyd-Graber & D. M. Blei, Syntactic Topic Models, `arXiv:1002.4665`, 2010.
5. J. R. Finkel, T. Grenager & C. D. Manning, The infinite tree, *Proceedings of the ACL*, 272–279, 2007.
6. K. T. Frantzi, S. Ananiadou, J. Tsujii, The C-value/NC-value Method of Automatic Recognition for Multi-word Terms, *LNCS* 1513, 585–604, 1998.
7. D. Ginev *et al.*, An Architecture for Linguistic and Semantic Analysis on the arXMLiv Corpus, *GI Jahrestagung 2009*: 3162–3176.
8. A. González-Palomo, Sentido: an authoring environment for OMDOC, in [Koh].
9. J. Gow & P. Cairns, Closing the Gap Between Formal and Digital Libraries of Mathematics, *Studies in Logic, Grammar and Rhetoric* 10 (23):249-263, 2007.

10. M. A. Hearst, Untangling Text Data Mining, *Proc. 37th Annual ACL Meeting*, 1999, p. 3–10.
11. Philipp Koehn, *Statistical Machine Translation*, Cambridge University Press, 2010.
12. F. Kamareddine *et al.*, Restoring Natural Language as a Computerised Mathematics Input Method, in *Calculemus '07/MKM '07, Proceedings of the 14th Symposium Towards Mechanized Mathematical Assistants*, Springer, 2007.
13. F. Kamareddine *et al.*, Narrative Structure of Mathematical Texts, in [12].
14. M. Kohlhase, *OAF: Flexiforms*, online.
15. M. Kohlhase, An Open Markup Format for Mathematical Documents, *LNAI* 4180, 2007.
16. C. Lange *et al.*, Reimplementing the MSC as a Linked Open Dataset, *CICM 2012*, LNAI 7362, 458–462.
17. C. Lange, *Enabling Collaboration on Semiformal Mathematical Knowledge by Semantic Web Integration*, PhD Thesis, Bremen, 2011.
18. W. Li & A. McCallum, Pachinko Allocation: DAG-Structured Mixture Models of Topic Correlations, *23rd Conference on Machine Learning*, 577–584, 2006.
19. E.D. Liddy, Natural Language Processing, in *Encyclopedia of Library and Information Science*, Marcel Decker, 2001.
20. R. Nallapati, A. Ahmed, E. P. Xing, W. W. Cohen, Joint Latent Topic Models for Text and Citations, *KDD'08*, 542–550, 2008.
21. G. Nenadić, S. Ananiadou & J. McNaught, Enhancing automatic term recognition through recognition of variation, *COLING 2004*, 604–610, 2004.
22. P. Quaresma, A XML-Format for Conjectures in Geometry, *CICM 2012 Conference*, work-in-progress section.
23. T. V. Raman, An audio view of LaTeX documents, *TUGboat* 13:3, 372–379, 1992 and 16:3, 310–314, 1995.
24. P. Rudnicki, An overview of the Mizar Project, in *Proc. of the 1992 Workshop on Types for Proofs and Programs*, 1992.
25. S. M. Watt, Mathematical Document Classification via Symbol Frequency Analysis, *DML 2008*, 29–40, 2008.
26. M. Wolska, M. Grigore & M. Kohlhase, Using Discourse Context to Interpret Object-Denoting Mathematical Expressions, *DML 2011*, 85–101, 2004.
27. C. W. Zinn, *Understanding Informal Mathematical Discourse*, PhD, Erlangen-Nürnberg, 2004.