Text-mining and machine-learning solid-state synthesis from the scientific literature

by

Haoyan Huo

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering- Materials Science and Engineering

and the Designated Emphasis

in

Computational and Data Science and Engineering

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Gerbrand Ceder, Chair
Professor Marti A. Hearst
Professor Daryl C. Chrzan
Dr. Anubhav Jain

Spring 2022

Text-mining and machine-learning solid-state synthesis from the scientific literature

Abstract

Text-mining and machine-learning solid-state synthesis from the scientific literature

by

Haoyan Huo

Doctor of Philosophy in Engineering- Materials Science and Engineering

and the Designated Emphasis in

Computational and Data Science and Engineering

University of California, Berkeley

Professor Gerbrand Ceder, Chair

Innovations of novel materials often involve synthesizing new compounds with better materials properties. However, computationally designing synthesis methods for these new compounds remains an uncharted new area of research. This thesis proposes to use machine-learning approaches to predict materials synthesis routes by training on synthesis information from the published scientific literature. However, most inorganic materials synthesis information in the scientific literature is locked-up in written natural language and must be parsed using natural language processing and information retrieval techniques. Therefore, this thesis aims to achieve two objectives: 1) constructing a text-mining pipeline that extracts solid-state synthesis datasets from scientific papers, and 2) implementing an interpretable machine-learning method to predict solid-state synthesis conditions.

Training information retrieval systems usually requires large manually labeled datasets, which are not widely available in materials informatics. To alleviate the lack of labeled datasets, we demonstrate a semi-supervised machine-learning method (Chapter 3), which is implemented for the classification of paragraphs in papers. Without any human labeling efforts, latent Dirichlet allocation can cluster keywords into topics corresponding to specific experimental synthesis steps. Guided by a small amount of annotation, supervised training methods, such as random forest, can then associate these steps with different synthesis methods, such as solid-state or hydrothermal synthesis. Using the topic modeling results, we also show a Markov chain representation of the order of experimental steps, which reconstructs a flowchart of synthesis procedures.

To fulfill the first objective, we have extracted a dataset of “codified recipes” for solid-state synthesis using an automated text-mining pipeline (Chapter 4). The dataset currently consists of over 30,000 solid-state synthesis entries. Every entry contains synthesis information including input materials, target materials, experimental operations, the associated processing parameters and synthesis conditions, and the balanced synthesis reaction equation. This dataset is the first-ever collection of machine-readable solid-state synthesis experiments and enables data mining of various aspects of inorganic materials synthesis.

To fulfill the second objective, we have built a machine-learning approach that predicts solid-state synthesis conditions (heating temperature and heating time) using the abovementioned dataset (Chapter 5). We used dominance importance ranking analysis and discovered that optimal heating temperatures have strong correlations with the stability of precursor materials. This correlation extends Tamman's rule from intermetallics to oxide systems, suggesting the importance of reaction kinetics in solid-state synthesis. Heating times are shown to be strongly correlated with the chosen experimental procedures and instrument setups, which may be indicative of the selection bias in the dataset. Our machine-learning models achieve good synthesis prediction performance and general applicability for diverse chemical systems.

While focusing particularly on solid-state synthesis, this thesis demonstrates a scalable framework to unlock the large amount of inorganic materials synthesis information from the literature, and machine-learn robust and interpretable synthesis predictors. At the end of this thesis, we outline several interesting future research topics which expand the work into a broader context of materials informatics and synthesis science.

This thesis is dedicated to my wife, Runzhi,
for her unconditional love and support.

Contents

List of Figures

List of Tables

Acknowledgments

During my years as a Ph.D. student, I was supported by many people at UC Berkeley and Lawrence Berkeley National Lab. It's hard to list all of them, but I am very grateful to all the people who had helped me through the years.

I'm greatly indebted to my advisor Gerbrand Ceder for his support and guidance, without which this thesis couldn't have been possible. He not only taught me how to think big for science problems but also helped me develop solid and practical skillsets to succeed in an academic environment.

I thank all my thesis committee members, specifically Prof. Gerbrand Ceder, Prof. Marti Hearst, Prof. Daryl Chrzan, and Dr. Anubhav Jain, for all their invaluable help for my research.

I thank my labmates in our text-mining subgroup. These names include Olga Kononova, Ziqin Rong, Tanjin He, Wenhao Sun, Vahe Tshitoyan, Tiago Botari, Chris Bartel, and Amalie Trewartha. It was so much fun working in this subgroup, and it is fascinating to see how far we have gone.

I thank the UC library who helped us negotiate publisher text-mining agreements that permitted us to work on the project. I also thank our collaborators at the MIT Department of Materials Science & Engineering, particularly Prof. Elsa Olivetti and Eddie Kim, for their engaging discussions and inspirations on this project.

In addition, I want to thank all the friends in the Ceder Group for all the wonderful memories.

Finally, I would like to thank my family for their support and encouragement when undertaking my research and working towards my degree. I couldn't have completed my Ph.D. without them.

Chapter 1

Introduction

1.1 Machine-learning approaches to solid-state synthesis design

Developing materials with good properties has both scientific and economic value. For example, information technology relies on the continuous innovation of many materials such as semiconductors, dielectric, magnetic, and optical materials; the goal of carbon neutrality depends on breakthroughs in energy storage and transportation technologies. Today, the design of new materials usually involves the **prediction** and **synthesis** of new inorganic compounds. In the last thirty years, the *prediction* of materials has been greatly accelerated by using a high-throughput computational pipeline [1], resulting in dozens of computationally designed novel compounds [2–6], and on-demand availability of *ab initio* predicted properties [7–11]. These computational methods typically search a huge list of hypothetical materials and identify the compounds with both good stability (to be stable and not decompose [12, 13]) and better material property (to deliver better application-specific performance).

However, the materials design pipeline remains bottlenecked by the challenges of *experimental synthesis*. To date, experimental synthesis is still mostly driven by tedious and laborious trial-and-error, as there is no comprehensive theory or model for designing synthesis routes. For example, the syntheses of many theoretically predicted materials with good properties failed [14–16] or remained very difficult [17, 18]. Understanding why syntheses are successful/failed for certain compounds, and then constructing models that guide future experimental syntheses, are therefore crucial steps toward enabling the next generation of accelerated materials development [19, 20].

The main subject of this thesis is conventional high-temperature solid-state synthesis [21] out of many possible inorganic materials synthesis methods. In its simplest form, solid-state synthesis requires mixing different input materials together and firing them under high temperatures (such as $1000\,^{\circ}\mathrm{C}$) such that the atoms can diffuse and form the target phases. Solid-state synthesis is the prevailing approach for making inorganic solids, but little of the reaction mechanism has been understood [22, 23]. Recently, there have been many works focusing on understanding solid-state reaction pathways using *in-situ* experiments [24–31], where the normally black-box reactions are decomposed into a sequence of phase evolution steps [22] that can be modeled using thermodynamic calculations (e.g., from first-principles) [32–36]. However, one of the biggest challenges yet to be solved is the prediction of solid-state synthesis conditions. To design synthesis routes for new materials, it is essential to understand why certain conditions are preferred and develop models for predicting the right conditions for synthesis. While thermodynamic calculations have been used to rationalize synthesis conditions in specific chemical systems [32, 37, 38], a synthesis condition predictor with broad applicability for general inorganic compounds is still elusive.

In this thesis, we propose and demonstrate machine-learning (ML) approaches in understanding and designing solid-state synthesis. The idea of using ML to solve complex design problems is no coincidence in the scientific community. Recently, ML has been used to intelligently design retrosynthesis routes in organic chemistry [39–43] and predict protein structures in biology [44, 45]. These ML investigations have been enabled by large-scale organic chemistry reaction databases [46, 47] and protein data banks [48, 49] where thousands, if not millions, of data points are stored in a machine-readable format. There is currently no analogous database that comprehensively catalogs the synthesis reactions of inorganic materials syntheses. However, the potential of ML synthesis design has been demonstrated by collecting and curating datasets in specialized domains [50, 51].

1.2 Motivating the application of text-mining and NLP for material synthesis

In this thesis, we aim to develop a solid-state synthesis predictor that works *universally for many compounds* rather than focusing on particular compositions. The core need to fulfill this objective is the abundance of synthesis datasets that cover many chemistry systems. Unfortunately, such experimental synthesis datasets have not been widely available. Many existing databases do contain experimental synthesis data which are manually constructed and curated over decades. For example, the Inorganic Crystal Structure Database (ICSD) [52] contains wide coverage of inorganic compounds that are reported experimentally; NIST Webbook [53], the Pauling File, and its derivative - Pearson's Crystal Data [54, 55] catalog synthesis and characterization results for intermetallics, oxides, and more. However, these databases either do not contain the experimental procedures at all (such as ICSD and NIST Webbook), or only contain semi-structured synthesis descriptions in natural language form (such as the Pauling File and Pearson's Crystal Data).

Journal articles may be the largest accessible collection of information on past synthesis experiments. The steady growth in the scientific community and the development of the Internet have led to a growing number of papers [56]. Today, it is estimated that millions of papers are published each year [57]. Indeed, our analysis of the papers indexed in the Web of Science repository shows that since the beginning of the 2000s, the number of publications in different fields of materials science has increased exponentially (Fig. 1.1).

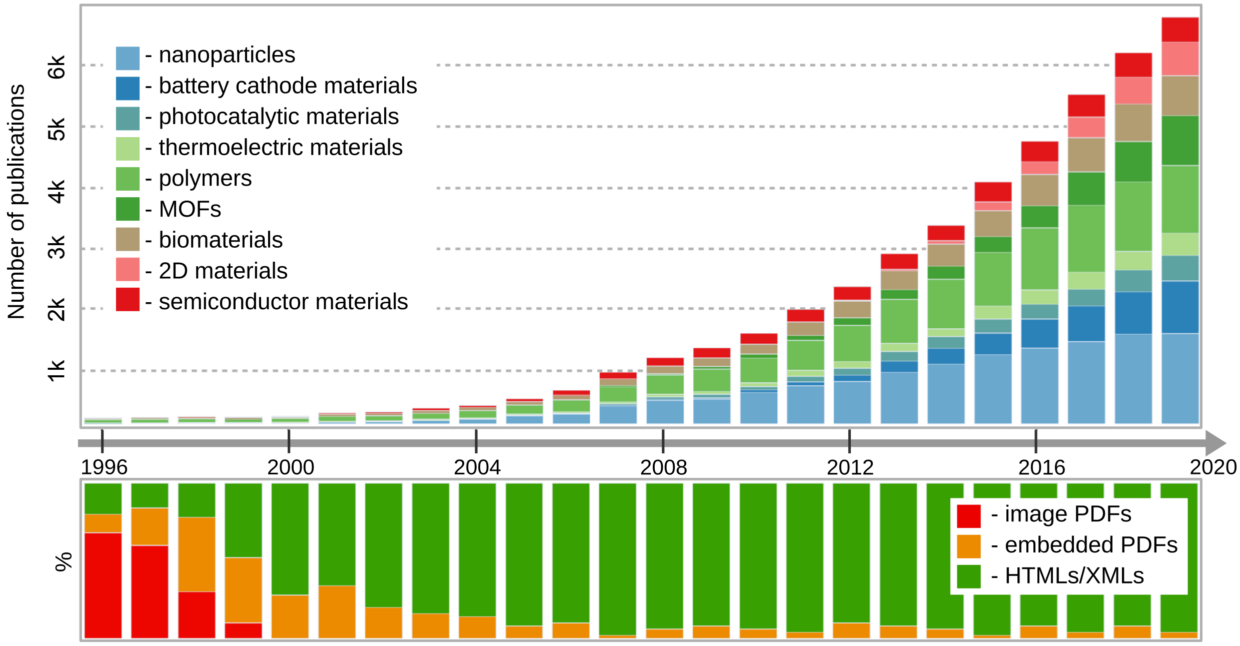


Figure 1.1: Publication trend in materials science over the past 14 years. Top panel: Number of publications appearing every year in different fields of materials science. All data were obtained by manually querying Web of Science publications resources. The analysis includes only research articles, communications, letters, and conference proceedings. The number of publications is on the order of 10^3. Bottom panel: Relative comparison of the fraction of scientific papers available online as image PDF or embedded PDF versus articles in HTML/XML format. The gray arrow marks time intervals for both top and bottom panels.

The development of text-mining and natural language processing (NLP) approaches has made it possible to implement various automated methodologies for converting scientific text into structured data collections [58]. For example, there have been a number of useful NLP toolkits for chemical text processing and information extraction, such as ChemDataExtractor [59], OSCAR4 [60], ChemicalTagger [61] and others [58, 62]. In more recent years, the potential of text-mining materials synthesis information has been demonstrated by the pioneering works by Kim *et al.* [63]. However, we note that no large-scale codified datasets on inorganic materials synthesis procedures were available, which is the first primary goal pursued by this thesis.

The information to be text-mined in this thesis is demonstrated in Fig. 1.2. To build a comprehensive inorganic materials synthesis database, synthesis information must be classified with high-resolution at multiple levels: at a high-level, the synthesis methodology; at an intermediate-level, individual experimental steps; and at a detailed-level, specific processing parameters. The detailed breakdown analysis of the techniques for text-mining of this information will be discussed in Chapter 2, while the implementations will be discussed in Chapter 4.

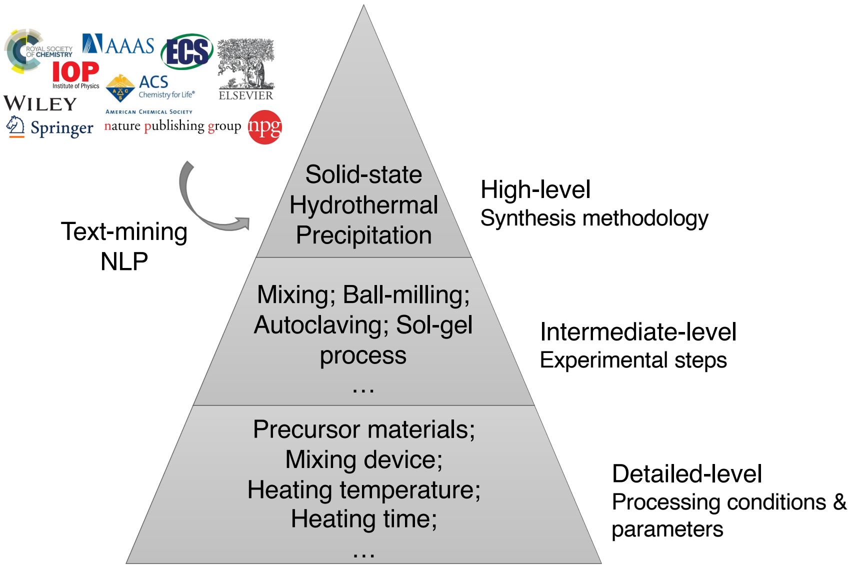


Figure 1.2: Objectives of text-mining and NLP for materials science journal articles.

1.3 Structure of this thesis

The work presented in this thesis aims at solving two problems: 1) developing an NLP infrastructure that text-mines synthesis information from journal articles, and 2) using the text-mined dataset to predict optimal synthesis conditions for any given reaction.

In Chapter 2, we review relevant works in the field of information retrieval (IR) and NLP, with the objective to adapt useful algorithms and tools into the materials science domain. Chapter 3 describes a semi-supervised approach of modeling synthesis procedure texts and classifying rare synthesis paragraphs in the corpus. The algorithms developed in Chapter 3 filters out the paragraphs in journal articles that do not contain synthesis reactions. This is important because significant more number of paragraphs that do not contain synthesis reactions would generate too many false positives for downstream named-entity recognition (NER) and parsing models. Implementing such paragraph filtering algorithms helps improve the overall extraction accuracy. The output of Chapter 3, a list of synthesis paragraphs that describe solid-state synthesis in a uniform language, were used as inputs for Chapter 4, where we developed a text-mining pipeline that produces a codified "recipes" dataset on solid-state synthesis. This codified dataset of "recipes" contains multiple aspects of solid-state synthesis and made it possible to perform data analysis and ML for synthesis prediction. Using this dataset, augmented with additional data such as reaction thermodynamics and materials properties, Chapter 5 describes an interpretable ML approach for rationalizing and predicting two important solid-state synthesis conditions, heating temperature and time. Finally, Chapter 6 summarizes the work in this thesis and outlines future works.

Chapter 2

Roadmap to text-mining materials science literature

Modern computers encode text as a sequence of bits representing each character, but without reflecting its internal structure or other high-order organization (e.g. words, sentences, paragraphs). Building algorithms to interpret the sequences of characters and to derive logical information from them is the primary purpose of text-mining and NLP. Unlike standard texts on general topics, such as newswire or popular press, scientific documents are written in specific language requiring sufficient domain knowledge to follow the ideas. Application of general-purpose text-mining and NLP approaches to the chemical or materials science domain therefore requires adaptation of both methods and models, including development of adequate training sets that comply with the goals of the text-mining project.

In this Chapter [1], we will review methods that are widely applied in text-mining scientific literature, with a heavy focus on adapting these methods into chemistry and/or materials science. General-purpose text-mining and NLP are rapidly evolving fields and beyond the scope of this Chapter. For curious readers, we also recommend other sources, such as the books by Miner *et al.* [65] and Jurafsky [66] as well as emerging papers and reviews in relevant NLP/IR fields.

2.1 Acquiring the text corpus

In computational linguistics, a large organized set of human-created documents is referred to as a *text corpus*. Scientific discourse generally occurs across a wide variety of document formats and types: abstracts in proceedings, research articles, technical reports and preprints, patents, e-encyclopedias, and many more. There are two primary ways to obtain

[1] This Chapter is based on parts of the previously published paper by Olga Kononova, Tanjin He, Haoyan Huo, Amalie Trewartha, Elsa A. Olivetti, and Gerbrand Ceder. “Opportunities and challenges of text mining in materials research.” *iScience*, Volume 24, Issue 3 (2021) [64] with permission from the authors.

the text corpus: i) by using existing indexed repositories with the available text-mining application programming interfaces (API) and search tools; or ii) by having access to individual publishers' content.

Text databases. A comprehensive overview of scientific text resources can be found in the review by Kolárik *et al.* [67]. Table 2.1 lists some common repositories for scientific texts in the domain of chemistry and material science, their document types, and access options. The main advantage of using established databases for text mining is the uniform format of their metadata, convenient API access, and sometimes established analysis tools. However, the majority of the publications in these repositories are heavily biased toward biomedical and biochemical subjects with a smaller fraction belonging to physics, (in)organic chemistry, and materials science. Moreover, the access to the content is limited: it either requires having a subscription or provides a search over open-access publications only.

Data Repository	**Documents Types**	**Access**	**Reference**
CAplus	research articles, patents, reports	subscription	www.cas.org/support/ documentation/references
DOAJ	research articles (open-access only)	public	doaj.org
PubMed Central	research articles	public	www.ncbi.nlm.nih.gov/pmc
Science Direct (Elsevier)	research articles	subscription	dev.elsevier.com/api_docs.html
Scopus (Elsevier)	abstracts	public	dev.elsevier.com/api_docs.html
Springer Nature	research articles, books chapters	subscription	dev.springernature.com/

Table 2.1: List of the some common text repositories in chemistry and material science subjects that provide an API for querying. *Note 1*: Elsevier provides API for both Science Direct (collection of Elsevier published full-text) and Scopus (collection of abstracts from various publishers). *Note 2*: Springer Nature provides access only to its own published full texts.

Individual publisher access. Implementation of customized scraping routines to screen the publisher's web-pages and download the content requires more effort. However, this approach allows for accessing content from those resources that are not providing an API, for example, e-print repositories. In most cases, downloading and accessing significant publisher content require text and data mining agreements. We note that this agreement differs from a standard academic subscription granted to the libraries of the institutions, because scraping and downloading large volumes affect the operation of the publishers' server.

Web-scraping not only requires a substantial amount of work, but also has to deal with dynamic web pages in which contents are generated by a client browser. In the work that will be described in Chapter 4, we implemented such a solution for Elsevier, the Royal Society of Chemistry, the Electrochemical Society, and American Institute of Physics publishers[68]. Similarly, ChemDataExtractor[59] provides the web-scrapers for Elsevier, RSC and Springer. In the research fields where most of the literature has an open access repository, e.g. physics, mathematics or the rapidly growing literature collection on COVID-19[69], corpus acquisition will be considerably easier.

2.2 Parsing structured plain text from documents

In general, the retrieved content includes text paragraphs and other metadata, such as journal names, titles, authors, keywords, and others. Querying text databases such as those in Table 2.1, provides a structured output with raw text ready for processing and analysis. In contrast, web-scraped contents usually consist of markup files requiring additional steps to convert them into raw text. Nowadays, most of the text sources are provided as HTML/XML/JSON documents, whereas older papers are usually available as embedded or image PDFs.

While parsing HTML/XML markups can be performed with various programming tools, extraction of the plain text from PDF files is more laborious. Embedded PDFs usually have a block structure with the text arranged in columns and intermixed with tables, figures, and equations. This affects the accuracy of conversion and text sequence. Some work has been done attempting to recover a logical text structure from PDF-formatted scientific articles by utilizing rule-based[70] and ML[71,72] approaches. However, the accuracy of these models measured as F1-score is still below $\sim$80%. The authors' experience demonstrates that this can dramatically impact the final output of the extraction pipeline. Hence, the decision on whether to include PDF text strongly depends on the tasks that are being solved.

A great number of documents, in particular those published before the 1990s, are only available as an image PDF files. Conversion of these files into raw text requires advanced optical character recognition (OCR) tools. To the best of our knowledge, the currently available solutions still fail to provide high enough accuracy to reliably extract chemistry[73,74]. Oftentimes, interpretation errors in PDFs originate from subscripts in chemical formulas and equations, and from confusion between symbols and digits. Creating a rigorous parser for PDF articles, and especially an OCR tool for scientific text is an area of active research in the computer science and text mining community[75,76].

2.3 Text pre-processing, grammatical and morphological parsing

The raw documents proceed through normalization, segmentation, and grammar parsing. During this step, the text is split into logical constitutes (e.g. sentences) and *tokens* (e.g. words and phrases), that are used to build a grammatical structure of the text. Depending on the final goals, the text tokens may be normalized by *stemming* or *lemmatization* and processed through *part-of-speech (POS) tagging* and *dependencies parsing* to build the sentences structure. These are explained below.

Paragraph segmentation and sentence tokenization identify, respectively, the boundaries of the sentences and word phrases (tokens) in a text. In general, finding the start/end of a sentence segment requires recognition of certain symbolic markers, such as period ("."), question mark ("?"), and exclamation mark ("!"), which is usually performed with (un)supervised ML models[77]. State-of-the-art implementations attain ∼95-98% accuracy (measured as F1-score). However, applying these models to scientific text requires modification. Commonly used expressions such as "Fig. X", "et al." and periods in chemical formulas often result in over-segmentation of a paragraph. Conversely, citation numbers at the end of a sentence promote the merging of two sentences together. There is no generally accepted solution to this problem, and it is usually approached by hard-coding a set of rules that capture particular cases[78].

Sentence tokenization, i.e. splitting a sentence into logical constituents, is a crucial step on the way to information extraction, because the errors produced in this step tend to propagate down the pipeline and affect the accuracy of the final results. Tokenization requires both unambiguous definition of grammatical tokens and robust algorithms for identification of the token boundaries. For general-purpose text, tokenization has been the subject of extensive research resulting in the development of various advanced methods and techniques[66]. However, for chemical and materials science text, accurate tokenization still requires substantial workarounds and revision of the standard approaches. Table 2.2 displays some typical examples of sentence tokenization produced by general-purpose tokenizers such as NLTK[79] and SpaCy[80]. As in the case of sentence segmentation, the major source of errors is the arbitrary usage of punctuation symbols within chemical formulas and other domain-specific terms. The chemical NLP toolkits such as OSCAR4[60], ChemicalTagger[61], and ChemDataExtractor[59] solve this problem by implementing their own rules- and dictionaries-based approaches to solve the over-tokenization problem. The advantage of chemical NLP toolkits is that they provide good performance on chemical terms, even if the rest of the text may have lower tokenization accuracy.

However, another prominent reason for tokenization errors is the lack of generally accepted rules regarding tokenization of chemical terms consisting of multiple words

<table>
<tr><td colspan="2"><i>Reagents (NH4)2HPO4 and Sm2O3 were mixed</i></td></tr>
<tr><td>NLTK</td><td>Reagents | <b>(</b> | <b>NH4</b> | <b>)</b> | <b>2HPO4</b> | and | Sm2O3 | were | mixed</td></tr>
<tr><td>SpaCy</td><td>Reagents | <b>(</b> | <b>NH4)2HPO4</b> | and | Sm2O3 | were | mixed</td></tr>
<tr><td>OSCAR4</td><td>Reagents | <b>(NH4)2HPO4</b> | and | Sm2O3 | were | mixed</td></tr>
<tr><td>ChemicalTagger</td><td>Reagents | <b>(NH4)2HPO4</b> | and | Sm2O3 | were | mixed</td></tr>
<tr><td>ChemDataExtractor</td><td>Reagents | <b>(NH4)2HPO4</b> | and | Sm2O3 | were | mixed</td></tr>
<tr><td colspan="2"><i>We made Eu2+-doped Ba3Ce(PO4)3 at 1200 °C for 2 h</i></td></tr>
<tr><td>NLTK</td><td>We | made | <b>Eu2+-doped</b> | <b>Ba3Ce</b> | <b>(</b> | <b>PO4</b> | <b>)</b> | <b>3</b> | at | 1200 | <b>°C</b> | for | 2 | h</td></tr>
<tr><td>SpaCy</td><td>We | made | <b>Eu2</b> | <b>+</b> | <b>-doped</b> | <b>Ba3Ce(PO4)3</b> | at | 1200 | ° | <b>C</b> | for | 2 | h</td></tr>
<tr><td>OSCAR4</td><td>We | made | <b>Eu2+</b> | <b>-</b> | <b>doped</b> | <b>Ba3Ce(PO4)3</b>| at | 1200 | <b>°C</b> | for | 2 | h</td></tr>
<tr><td>ChemicalTagger</td><td>We | made | <b>Eu2+-doped</b> | <b>Ba3Ce(PO4)3</b> | at | 1200 | <b>°C</b> | for | 2 | h</td></tr>
<tr><td>ChemDataExtractor</td><td>We | made | <b>Eu2+</b> | <b>-</b> | <b>doped</b> | <b>Ba3Ce(PO4)3</b> | at | 1200 | ° | <b>C</b> | for | 2 | h</td></tr>
<tr><td colspan="2"><i>Lead-free a(Bi0.5Na0.5)TiO3-bBaTiO3-c(Bi0.5K0.5)TiO3 ceramics were investigated</i></td></tr>
<tr><td>NLTK</td><td><b>Lead-free</b> | <b>a</b> | <b>(</b> | <b>Bi0.5Na0.5</b> | <b>)</b> | <b>TiO3-bBaTiO3-c</b> | <b>(</b> | <b>Bi0.5K0.5</b> | <b>)</b> | <b>TiO3</b> | ceramics | was | investigated</td></tr>
<tr><td>SpaCy</td><td><b>Lead</b> | <b>-</b> | <b>free</b> | <b>a(Bi0.5Na0.5)TiO3-bBaTiO3-c(Bi0.5K0.5)TiO3</b> | ceramics | was | investigated</td></tr>
<tr><td>OSCAR4</td><td><b>Lead</b> | <b>-</b> | <b>free</b> | <b>a(Bi0.5Na0.5)TiO3-bBaTiO3-c(Bi0.5K0.5)TiO3</b> | ceramics | was | investigated</td></tr>
<tr><td>ChemicalTagger</td><td><b>Lead-free</b> | <b>a(Bi0.5Na0.5)TiO3-bBaTiO3-c(Bi0.5K0.5)TiO3</b> | ceramics | was | investigated</td></tr>
<tr><td>ChemDataExtractor</td><td><b>Lead-free</b> | <b>a(Bi0.5Na0.5)TiO3-bBaTiO3-c(Bi0.5K0.5)TiO3</b> | ceramics | was | investigated</td></tr>
</table>

Table 2.2: Examples of how different tokenizers split sentences into tokens. NLTK[79] and SpaCy[80] are general-purpose tokenizing tools, whereas ChemDataExtractor[59], OSCAR4[60], ChemicalTagger[61] are the tools trained for a scientific corpus. Tokens are bound by "|" symbol.

For instance, complex terms such as "lithium battery" or "yttria-doped zirconium oxide" or "$(Na_{0.5}K_{0.5})NbO_3$ + x wt% CuF_2" often become split into separate tokens "lithium" and "battery", "yttria-doped" and "zirconium" and "oxide", "$(Na_{0.5}K_{0.5})NbO_3$" and "+" and "x wt% CuF_2". This significantly modifies the meaning of the tokens and usually results in lowered accuracy of the named entity recognition (see below). Currently, this problem is solved case-by-case by creating task-specific wrappers for existing tokenizers and named entity recognition models[81–83]. Building a robust approach for chemistry-specific sentence tokenization and data extraction requires a thorough development of standard nomenclature for complex chemical terms and materials names.

More recent works in NLP have started to model tokenization as an character encoding task rather than a dictionary look-up task. Tokenizers such as Byte-Pair-Encoding (BPE) [84], WordPiece [85], and SentencePiece [86] use unsupervised methods to look for the most efficient encoding using a list of *subwords*, which are the most frequently used parts in a specific corpus. In such tokenizers, a single word may be divided into many subwords, e.g., “processing” → “process” and “ing”, since subword “ing” appears frequently in English. This not only allows one to train corpus-aware tokenization models without manually annotating the text, but also ensures the tokenization is optimized for specific corpora, potentially accounting for rare words and punctuation, which is known to affect the performance of very large neural network language models [86]. Similarly, special words and notations in chemistry and materials science may be learned without any human supervision, which can be beneficial for downstream NLP tasks [87].

Text normalization, part-of-speech tagging, and dependency parsing are often used to reduce the overall document lexicon and to design words’ morphological and grammatical features used as an input for entity extraction and other text-mining tasks[78]. Text normalization usually consists of lemmatization and/or its simpler version – stemming. While during the stemming the inflected word is cut to its stem (e.g. “changed” becomes “chang”), lemmatization aims to identify a word’s lemma, i.e. a word’s dictionary (canonical) form (e.g. “changed” becomes “change”)[66]. Stemming and/or lemmatization help to reduce the variability of the language, but the decision whether to apply it or not, depends on the task and expected outcome. For instance, recognition of chemical terms will benefit less from stemming or lemmatization[88] as it may truncate a word’s ending resulting in a change of meaning (compare “methylation” vs. “methyl”). But when a word identifies, for example, a synthesis action, lemmatization helps to obtain the infinitive form of the verb and avoids redundancy in the document vocabulary[68].

POS tagging identifies grammatical properties of the words and labels them with the corresponding tags, i.e. noun, verb, article, adjective, and others. This procedure does not modify the text corpus but rather provides linguistic and grammar-based features of the words that are used as input for ML models. A challenge in identifying the POS tags in scientific text often arises due to the specialized usage of English words, which may not be common in day-to-day English. As an example, compare two phrases: “the chemical tube is on the ground” and “the chemical was finely ground”. In first case, the general-purpose POS tagger will work correctly, while in the second example, it will likely misidentify “ground” as adjective or nouns if the usage of “ground” as a verb is not frequent in training data. Therefore, using a standard POS tagger often requires re-training of the underlying NLP model weights, or post-processing and correction of the obtained results, to further improve the accuracy of POS tagging in scientific contexts.

Dependency parsing creates a mapping of a linear sequence of sentence tokens into a hierarchical structure by resolving the internal grammatical dependencies between the words. This hierarchy is usually represented as a *dependency tree*, starting from the *root* token and going down to the terminal nodes. Parsing grammatical dependencies helps to deal with the arbitrary order of the words in the sentence and establishes semantic relationships between words and parts of the sentence[66]. Grammatical dependency parsing is a rapidly developing area of NLP research providing a wealth of algorithms and models for general-purpose corpus (see www.nlpprogress.com for specific examples and evaluation).

Application of the currently existing dependency parsing models to scientific text comes with some challenges. First, sentences in science are often depersonalized, with excessive usage of passive and past verbs tense, and limited usage of pronouns. These features of the sentence are not well captured by general-purpose models. Secondly, the accuracy of the dependency tree construction is highly sensitive to punctuation and correct word forms, particularly verb tenses. As scientific articles do not always exhibit perfect language grammar, the standard dependency parsing models can produce highly unpredictable results. To the best of our knowledge, these specific challenges of dependency parsing for scientific text have not yet been addressed or explored in detail.

2.4 Text representation modeling and deep learning

The application of ML models requires mapping the document into a linear (vector) space. A common approach is to represent a text as a collection of multidimensional (and finite) numerical vectors that preserve the text features, e.g. synonymous words and phrases should have a similar vector representation, and phrases having an opposite meaning should be mapped into dissimilar vectors[89]. Modeling of the vectorized text representation is a broad and rapidly developing area of research[90]. In this section, we highlight only some of the approaches applied to scientific text-mining, whereas a more detailed discussion of the methods can be found elsewhere[66].

The *bag-of-words model* is one of the simplest models of text representation. It maps a document into a vector by counting how many times every word from a predefined vocabulary occurs in that document. While this model works well for recognizing specific topics defined by keywords, it does not reflect word context and cannot identify the importance of a particular word in the text. The latter can be solved by introducing a normalization factor and applying it to every word count. An example of such normalization is the *tf-idf model* (*term frequency-inverse document frequency*) which combines two metrics: the frequency of a word in a document and the fraction of the documents containing the word. The method can thereby identify the terms specific to a particular document. Bag-of-words and tf-idf are the most commonly used models to classify scientific documents or to identify parts of text with relevant information[63,91,92].

While bag-of-words and tf-idf are relatively versatile, they do not identify similarity between words across documents. This can be done through *topic modeling* approaches[93]. Topic modeling is a statistical model that examines the documents corpus and produces a set of abstract topics – clusters of the keywords that characterize a particular text. Then, every document is assigned with a probability distribution over topical clusters. Latent Dirichlet allocation (LDA), a specific topic modeling approach,[94] has been applied to analyze the topic distribution over materials science papers on oxide synthesis[63] and to classify these papers based by synthesis method used in the paper[95].

Significant progress in text-mining and NLP has been achieved with the introduction of *word embedding* models which construct a vectorized representation of a single word rather than of the entire document. These approaches use the distributional hypothesis[89] and are based on neural networks trained to predict word context in a self-supervised fashion. Multiple variations of word embeddings models include GloVe[96], ELMo[97], word2vec[98] and FastText[99]. Besides being intuitively simple, the main advantage of word embedding models is their ability to capture similarity and relations between words based on mutual associations. Word embeddings are applied ubiquitously in materials science text-mining and NLP to engineer words features that are used as an input in various named entity recognition tasks[68,81,100,101]. Moreover, they also seem to be a promising tool to discover properties of materials through words association[102].

Recently, research on text representation has shifted toward context-aware models. A breakthrough was achieved with the development of *sequence-to-sequence models*[103] and, later, an *attention mechanism*[104] for the purpose of neural machine translation. The most recent models such as Bidirectional Encoder Representations from Transformers (BERT)[105] and Generative Pre-trained Transformer (GPT)[106,107] are multi-layered deep neural networks trained on very large unlabeled text corpora, and demonstrate state-of-the-art NLP performance. These models offer fascinating opportunities for the future NLP development in domain of materials science[87,108,109].

2.5 Information retrieval from the text

IR represents a broad spectrum of NLP tasks that extract various types of data from the pre-processed corpus. The most ubiquitous IR task is *NER* which classifies text tokens in a specific category. In general-purpose text, these categories are usually names of locations, persons, etc., but in scientific literature the named entities can include chemical terms as well as physical parameters and properties. Extraction of action graphs of chemical synthesis and materials fabrication is another class of IR task that is closely related to NER. This task requires identification of action keywords, linking of them into a graph structure, and, if necessary, augmenting with the corresponding attributes characterizing the action (e.g. the action "material mixing" can be augmented with the attribute "mixing media" or "mixing time"). Lastly, data extraction from figures and tables represents another class of information that can be retrieved from scientific literature. This requires not only text-mining methods but also image recognition approaches. In this section we will mainly review the recent progress for chemical and materials NER and action graphs extraction, and will provide a brief survey of the efforts spent on mining of scientific tables and figures.

Chemical NER is a broadly defined IR task. It usually includes identification of chemical and materials terms in the text, but can also involve extraction of properties, physical characteristics and synthesis actions. The early applications of chemical NER were mainly focused on extraction of drugs and biochemical information to perform more effective document searches[60,88,110,111]. Recently, chemical NER has shifted towards (in)organic materials and their characteristics[59,83,101,112], polymers[113], nanoparticles[92], synthesis actions and conditions[61,63,68,109]. The methods used for NER vary from traditional rule-based and dictionary look-up approaches to modern methodology built around advanced ML and NLP techniques, including conditional random field (CRF)[114], long short-term memory (LSTM) neural networks[115], and others. A detailed survey on the chemical NER and its methods can be found in recent reviews[58,116,117].

Extraction of chemical and materials terms has been a direction of intensive development in the past decade[58,62]. The publicly available toolkits use rules- and dictionaries-based approaches (e.g LeadMine[118]), statistical models (e.g OSCAR4[60]), and, predominantly, the CRF model (e.g. ChemDataExtractor[59], ChemSpot[110], tmChem[78]) to assign labels to chemical terms. Some recent works implemented advanced ML models such as bi-directional LSTM models[83,101,108] as well as a combination of deep convolutional and recurrent neural networks[119] to identify chemical and material terms in the text and use context information to assign their roles. Table 2.3 shows a few examples of the NER output obtained using some of these tools and compares it to non-scientific NER models implemented in NLTK[79] and SpaCy[80] libraries.

Often, the objective of scientific NER task is not limited to the identification of chemicals and materials, but also includes recognition of their associated attributes: structure and properties, amounts, roles and actions performed on them. Assigning attributes to the entities is usually accomplished by constructing a graph-like structure that links together all the entities and build relations between them. A commonly used graph structure is the grammatical dependency tree for a sentence (see Section 2.3). Traversing the sentence trees allows for resolving relations between tokens, hence, link the entities with attributes. ChemicalTagger[61] is one of the most robust frameworks that extends the OSCAR4[60] functionality and provides tools for grammatical parsing of chemical text to find the relation between entities and the corresponding action verbs. Similarly, ChemDataExtractor[59] can identify the chemical and physical characteristics (e.g. melting temperature) in the text and assign it to a material entity. A rules- and dictionaries-based relation-aware

An aqueous solution was prepared by dissolving lithium, cobalt, and manganese nitrates in de-ionized water	
NLTK	–
SpaCy	‘manganese’ (*nationalities or religious or political groups*)
OSCAR4	‘aqueous’, ‘lithium’, ‘cobalt’, ‘manganese’, ‘nitrates’, ‘water’
tmChem	‘lithium’, ‘cobalt’, ‘manganese nitrates’
ChemDataExtractor	‘lithium’, ‘cobalt’, ‘manganese nitrates’
ChemSpot	‘lithium’, ‘cobalt’, ‘manganese nitrates’, ‘water’
Bi-LSTM ChNER	‘lithium, cobalt, and manganese nitrates’, ‘water’
A series of Ce3+-Eu2+ co-doped Ca2Si5N8 phosphors were successfully synthesized	
NLTK	–
SpaCy	–
OSCAR4	‘Ce3+’, ‘Eu2+’, ‘Ca2Si5N8’
tmChem	‘Ce3+-Eu2+’, ‘Ca2Si5N8’
ChemDataExtractor	‘Ce3+-Eu2+’, ‘Ca2Si5N8’
ChemSpot	‘Ce3+-Eu2’, ‘co’, ‘Ca2Si5N8’
Bi-LSTM ChNER	‘Ce3+-Eu2+ co-doped Ca2Si5N8’
High-purity Bi(NO3)3·5H2O, Ni(NO3)2·6H2O and Cu(CH3COO)2·H2O were used as starting materials for Bi2Cu1-xNixO4 powders	
NLTK	‘NO3’, ‘NO3’, ‘CH3COO’ (*organizations*); ‘Ni’, ‘Cu’ (*countries, cities, states*)
SpaCy	‘Bi2Cu1-xNixO4’ (*person*)
OSCAR4	‘Bi(NO3)3·5H2O’, ‘Ni(NO3)2·6H2O’, ‘Cu(CH3COO)2·H2O’
tmChem	‘Bi(NO3)3·5H2O’, ‘Ni(NO3)2·6H2O’, ‘Cu(CH3COO)2·H2O’, ‘Bi2Cu1-xNixO4’
ChemDataExtractor	‘Bi(NO3)3·5H2O’, ‘Ni(NO3)2·6H2O’, ‘Cu(CH3COO)2·H2O’, ‘Bi2Cu1-xNixO4’
ChemSpot	‘Bi(NO3)3·5H2O’, ‘Ni(NO3)2·6H2O’, ‘Cu(CH3COO)2·H2O’, ‘Bi2Cu1-xNixO4’
Bi-LSTM ChNER	‘Bi(NO3)3·5H2O’, ‘Ni(NO3)2·6H2O’, ‘Cu(CH3COO)2·H2O’, ‘Bi2Cu1-xNixO4’

Table 2.3: Examples of the chemical named entities extracted by different NER tools. NLTK[79] and SpaCy[80], and the tools trained on chemical corpus OSCAR4[60], tmChem[78], ChemSpot[110], ChemDataExtractor[59], Bi-LSTM chemical NER[83]. For the general-purpose tools, the assigned labels are given in parenthesis. For the chemical NER, only entities labeled as chemical compounds are shown.

chemical NER model has been proposed by Shah *et al.* [112] to build a search engine for publications. Weston *et al.* [101] used the random forest decision model to resolve synonyms between chemical entities and materials-related terms. He *et al.* [83] applied a two-step LSTM model to resolve the role of materials in a synthesis procedure. Onishi *et al.* [120] used convolutional neural network model to build relations between materials, their mechanical properties and processing conditions which were extracted from publications by keywords search. Lastly, a combination of advanced NLP models has been recently used to extract the materials synthesis steps and link them into an action graph of synthesis procedures for solid-state battery materials[108] and inorganic materials in general[121].

Despite significant effort, the accuracy of the NER for chemical names and formulas is still relatively low compared to the general state-of-the-art NER models[122,123]. Fig. 2.1a displays the overall precision and recall for different chemical NER models reported in the corresponding publications. Both, precision and recall of the models vary from 60% to 98% (Fig. 2.1a), whereas for the general-purpose NER, these values are >91% (see www.nlpprogress.com). There are three major challenges that obstruct training of high-accuracy chemical NER models:

- The lack of unambiguous definitions of the chemical tokens and their boundaries.
- The lack of the robust annotation schema as well as comprehensive labeled training sets for the supervised ML algorithms.
- Higher grade level of scientific paragraphs compared to general English corpora.

Oftentimes, researchers manually create their own training set for specific tasks but with limited use for more general goals. Therefore, the success of chemical NER becomes a trade-off between the size of the annotated set and model complexity: either using simple model with limited capabilities on a small set of labeled data, or investing effort into annotation of a large dataset and using it with advanced models providing a higher accuracy of data extraction.

An early attempt in creating a labeled dataset for the chemical NER task was done by Kim *et al.* [125] and Krallinger *et al.* [126]. The GENIA and CHEMDNER sets provide annotation schema and labeled data of chemicals and drugs extracted from MEDLINE and PubMed abstracts, respectively. However, these corpora are heavily biased toward biomedicine and biochemical terms with only a small fraction of organic materials names present. The progress of the past few years brought a variety of annotated corpora to the materials science domain. Among the publicly available labeled dataset, there is the NaDev corpus consisting of 392 sentences and 2,870 terms on nanocrystal device development[127], the dataset of 622 wet lab protocols of biochemical experiments and solution syntheses[128],

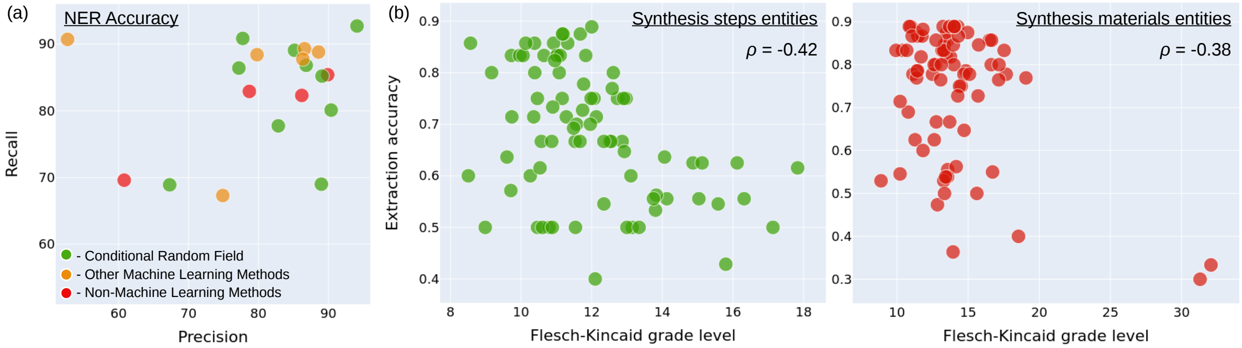


Figure 2.1: Performance of NER models in various materials synthesis text-mining applications. *Panel a:* Precision and recall of the published models for chemical NER manually extracted from the reports. Color denotes the primary algorithm underlying the model. *Panel b:* Accuracy of the data extracted from materials synthesis paragraphs plotted against the complexity of the paragraphs. The accuracy is computed using chemical NER models developed by our team[68,83] to the manually annotated paragraphs. The text complexity is calculated as a Flesch-Kincaid grade level (FKGL) score indicating the education level required to understand the paragraph[124]. ρ is a Pearson correlation coefficient between the accuracy of NER model and the FKGL score.

a set of 9,499 labeled sentences on solid oxide fuel cells[129], and an annotated set of 230 materials synthesis procedures[130].

Extraction of information from tables and figures is another branch of scientific IR that has been rapidly developing in the past few years. The specific format of the figures and tables in scientific papers imposes substantial challenges for the data retrieval process. First, it is common that images (and sometimes the tables) are not directly embedded in the HTML/XML text but instead contain a link to an external resource. Second, connecting tables/images to the specific part of the paper text is an advanced task that does not have a robust solution to date. Third, both tables and images can be very complex: images can include multiple panels and inserts that require segmentation, while tables may have combined several rows and columns imposing additional dependencies on the data. To the best of our knowledge, only a few publications have attempted to parse tables from the scientific literature using heuristics and machine learning approaches[131,132].

Image recognition methods have been broadly used in materials science but have so far been primarily focused on extracting information about the size, morphology, and the structure of materials from microscopy images. To date, the existing solutions for interpretation of microscopy images use variations of convolutional neural networks, and address diverse spectra of materials science problems[133–136]. While these models demonstrate a remarkable accuracy when applied directly to microscopy output, they are not intended to separate and process the images embedded in scientific articles. Steps toward parsing of article's images were reported recently. Mukaddem *et al.* [137] developed the ImageDataExtractor tool that uses a combination of OCR and CNN to extract the size and shape of the particles from microscopy images. Kim *et al.* [138] used Google Inception-V3 network[139] to create the Livermore SEM Image Tools for electron microscopy images. This tool was later applied by Hiszpanski *et al.* [92] on ∼35,000 publications to obtain information about the variability of nanoparticles sizes and morphologies.

2.6 Conclusions

In this Chapter, we have revisited the key components of a scientific literature text-mining pipeline. The infrastructure often starts with a text **corpus acquisition** part which collects thousands or millions of articles with as broad as possible topics and domains. A **document parser** aims to eliminate garbage contents by understanding document structure and extracting the hierarchy of text paragraphs. The domain knowledge starts to be injected into the pipeline when we develop chemistry- or materials-science-specific **pre-processors** and **NLP parsers**, which must be built with much consideration of the specific language systems used in the domain. A final component in the infrastructure often is the **NER** and **parsing** models that extract very specific items from the text. Building such systems is not an easy task as much of the existing works in NLP have to be adapted. These domain-specific adaptions are the essential contents in building a inorganic materials synthesis text-mining pipeline and will be discussed in the next Chapters.

Chapter 3

A semi-supervised approach of modeling synthesis procedures

Synthesis information is rare in journal articles. Journal publishers usually aim to publish contents for a broad audience, resulting in a diverse set of topics in the journal articles. Even though we limit the broad topics by manually choosing specific list of journal articles, we still found that only less than 5% of all papers in our database contain inorganic materials synthesis information. Within each paper, there are usually only $1 \sim 3$ synthesis paragraphs out of > 20 all the paragraphs. This imbalance of text contents creates challenges for IR systems and requires creating synthesis paragraph classifiers that prevent non-synthesis text from entering the text-mining pipeline.

In principle, we could analyze sentence grammar and keywords to build a rule-based classification algorithm to identify different types of synthesis procedures. However, this is impractical, due to both the notorious ambiguity of natural language [140–142] and the complexity of solid-state chemistry terminology. Statistical classification algorithms, such as deep-learning neural networks [143, 144] can achieve good text classification performances [58] with large amounts of training data [145]. However, no large annotated text data sets to train on exist in materials science or chemistry.

Recent advances in machine-learning have demonstrated that semi-supervised learning methods can solve similar classification problems with much less annotated data than supervised learning methods [146–148]. In this Chapter [1], we present a semi-supervised machine-learning approach (that uses a small amount of labeled data and a large amount of unlabeled data) for the accurate classification of synthesis procedures as

[1] This Chapter is based on the previously published paper by Haoyan Huo, Ziqin Rong, Olga Kononova, Wenhao Sun, Tiago Botari, Tanjin He, Vahe Tshitoyan, and Gerbrand Ceder. “Semi-supervised machine-learning classification of materials synthesis procedures.” *npj Computational Materials*, Volume 5, Issue 62 (2019) [95] with permission from the authors.

described in written natural language. Using a body of 2,284,577 articles, we applied LDA [94] to identify the experimental steps implied in sentences in an unsupervised manner. Without any human input, LDA can cluster keywords into topics corresponding to specific experimental materials synthesis steps, such as “grinding” and “heating”, “dissolving” and “centrifuging”, etc. The “experimental steps” are grouped as topics and LDA provides a probabilistic topic distribution for each sentence. To this topic distribution, we apply the random forest (RF) algorithm [149] - a supervised machine-learning method — to classify different types of synthesis procedures: solid-state synthesis, hydrothermal synthesis, sol–gel precursor synthesis, or none of the above. We demonstrate that the RF models can achieve high classification performance with training data sets as small as a few hundred paragraphs, which can be readily prepared by manual annotation efforts. By combining these unsupervised and supervised approaches, our machine-learning algorithm accurately captures the features and subtleties of different synthesis procedures, with high classification performance, with results that can be presented in a way that is readily understood and interpretable by humans. Finally, we construct a machine-learned flowchart of synthesis procedures, which demonstrates that our method can build a “machine intuition” of materials synthesis procedures beyond classification. The ideas and methods presented in this Chapter enable a scalable approach to unlock the large amount of inorganic materials synthesis information from the literature and to process it into a standardized, machine-readable database.

3.1 Unsupervised topic modeling (LDA) of synthesis processes

Humans can categorize sentences into topics by recognizing familiar keywords. However, this objective can be difficult to train a computer to achieve, because it is impractical to code explicit rules for keywords of an English vocabulary that is both large ($> 10,000$) and open for new materials science/chemistry terms. Furthermore, in natural language various synonyms can often be used to represent the same topic, which introduces ambiguity and complexity into hard-coded rules. LDA [93, 94, 150] is an unsupervised topic modeling algorithm that observes common keywords over a large number of papers, then automatically clusters these synonymous keywords together into “topics”. We applied LDA to identify topics of synthesis from the scientific literature, and we demonstrate that the topical grouping is closely related to conventional experimental classification of synthesis steps.

We first use LDA to identify topic–word distributions, which are a set of multinomial probability distributions over a cluster of keywords conditioned on certain topics. To demonstrate, in Table 3.2 we list two topics learned by LDA. We first show in Table 3.2 some representative sentences that we consider to discuss similar topics. From a collection of thousands of unlabeled sentences, LDA learns topic–word distributions using a Bayesian

Sample sentences	**Words of highest probability**	**Topics**
"As-received ZrB2 powder was mixed with 2 wt% B4C powder (4.5 vol%) and 1 wt% carbon (2.5 vol%) in acetone by ball milling for 24 h using WC media." [151]	P(ball) = 0.065 P(milling) = 0.051 P(h) = 0.042 P(milled) = 0.032 P(powder) = 0.031 ...	T_1 (ball-)milling
"The Al powder was first ball milled in an atmosphere of supra-pure hydrogen for removing the small amount of oxide film on the surface." [152]		
"The solid product obtained was filtered, dried at 110 °C and finally calcined in air at 550 °C for 6 h at a heating rate of 1 °C/min. "[153]	P(°C) = 0.139 P(h) = 0.104 P(air) = 0.038 P(calcined) = 0.035 P(dried) = 0.028 ...	T_2 sintering
"Finally, the solid was calcined in air from RT to 500 °C at a heating rate of 2 °C min-1 and maintained for 4 h, which led to the formation of the MgO-Al2O3 support." [154]		

Table 3.2: Two topics (topic–word distributions) selected from 200 topics learned by LDA using sentences in our database. Each topic is represented by a multinomial probability distribution over words. By interpreting the keywords (words of highest probability), we assign a human comprehensible label for each topic. Sample sentences from four articles [151–154] are used to demonstrate different topics.

inference method. As shown in the second column of Table 3.2, the keywords (words of highest probability) of topics match the vocabulary often used by chemists to discuss each topic, making it possible for chemists to interpret the learned topics. For example, in Table 3.2, we interpret topic T_1 as "(ball-)milling", and topic T_2 as "high temperature sintering". We emphasize that the topic names, "(ball-)milling" and "sintering", are assigned by us for the sake of convenience, and the choice of names does not affect the topic–word distributions learned by LDA.

The distribution of topics in a sentence infers a "document–topic" distribution, which is quantified by the probability that each topic appears in a sentence. For example, in a sentence excerpted from our database, "the dried powders were calcined twice at $850°\mathrm{C}$ for $2h$ and then ball milled again for $8h$." [155], 39 and 60% of the words discuss the LDA-learned topics T_1 and T_2, respectively. LDA then interprets this sentence as having two topics, corresponding to the experimental steps "ball milling" and "sintering". Using document–topic distributions, a computer is able to quantitatively identify topics relevant to experimental steps in sentences, which are then used as input features for synthesis procedure classifiers.

3.2 Supervised classification of synthesis methodologies

LDA has now been used to automatically identify various topic–word distributions, which we labeled as specific experimental steps, for example, sintering, grinding, etc. These individual steps are subprocesses of an overall synthesis methodology, such as solid-state synthesis, hydrothermal, sol–gel precursor synthesis, etc. Based on the topic distributions learned by LDA, the machine is next trained to classify which of these three synthesis methodologies a synthesis paragraph corresponds to.

To build the classifier, we use the RF algorithm [149, 156], which is a supervised machine-learning algorithm that uses an ensemble of decision-making trees to make classifications. We constructed a training set of synthesis paragraphs that was annotated by synthesis experts, which consists of 1000 training paragraphs for each of the three types of synthesis (solid-state, hydrothermal, and sol–gel precursor synthesis) as well as 3000 randomly sampled negative paragraphs from the database that do not contain any of the above three synthesis procedures. To provide input features for RF, we use the "topic n-gram" [66], which represents the sequence of LDA-derived topics in adjacent sentences within a paragraph, as demonstrated in Algorithm 1. In this work, we used 1-, 2-, and 3-sized "topic n-grams". The rationale can be understood by the fact that synthesis experiments consists of smaller *modules* made up by 1-3 steps in fixed order. These modules can be reordered to increase the complexity of experiments (i.e., repeated grinding and firing, synthesis of several intermediate compounds, etc.), thus achieving different goals of synthesis (i.e., different grain size, avoiding the formation of impurities, etc. We used the scikit-learn Python package [157] to construct learning curves to understand how much training data is needed by the RF algorithm.

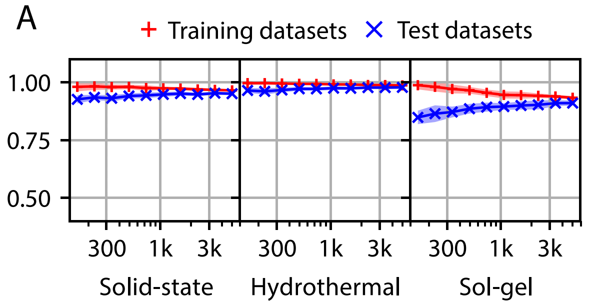
We evaluate the performance of our models by splitting a manually annotated dataset into 5000 training samples and 1000 hold-out samples. The hyperparameters are optimized using five-fold cross-validation on the training samples, and we report performance on the hold-out dataset. Figure 3.1a gives the learning curves of the RF algorithm, showing the F1 score versus the amount of training data. The RF algorithm reaches high F1 scores of $\sim 90\%$ when the training data set size is > 3000, but surprisingly, the models can consistently converge to $> 80\%$ F1 scores even when the training data set is as small as a few hundred paragraphs. These training data sets are small enough that they can be readily prepared by manual annotation efforts, indicating that LDA + RF methods are practicable machine-learning methods for classification problems of similar complexity. As summarized in Fig. 3.1b, the recall and precision scores are also $> 90\%$, signifying that our RF classification model is robust against false-positive and false-negative classification

```
**Data:** Sentences $s_i, \cdots, s_M$ in a paragraph.
**Data:** Topics $T_{i,j}$ for each sentence $s_i$ and their probability in a sentence $P_i(T_{i,j})$.
**Parameter:** Tuple size $N$, topic tuple joint probability threshold $P_0$
**Result:** List of topic n-gram $G$ and their numerical feature value $V$
$G \leftarrow \emptyset$;
$V \leftarrow [\,]$;
for $i = 1, 2, \cdots M$ do
    /* Iteration to generate N-sized topic tuples */
    for $t_i = T_{i,1}, T_{i,2}, \cdots$ do
        for $t_{i+1} = T_{i+1,1}, T_{i+1,2}, \cdots$ do
            $Repeat\ for\ loop\ N\ times \cdots$;
            for $t_{i+N-1} = T_{i+N-1,1}, T_{i+N-1,2}, \cdots$ do
                /* Filtering based on joint topic probability */
                $p \leftarrow P_i(t_i) \cdot P_{i+1}(t_{i+1}) \cdots \cdot P_{i+N-1}(t_{i+N-1})$;
                if $p > P_0$ then
                    $NGram \leftarrow \{t_i, t_{i+1}, \cdots, t_{i+N-1}\}$;
                    $G \leftarrow G \cup \{NGram\}$;
                    $V[NGram] \leftarrow V[NGram] + p$
                else
            end
        end
    end
end
```

Algorithm 1: Generation of "topic n-gram" features. Once G and V are constructed for all paragraphs, they are converted into a fixed size vector whose components correspond to each of the n-grams $g \in G$ and the values $v \in V$. V measures the strength of each topic n-gram and is derived from its joint probabilities.

errors.

The RF algorithm consists of an ensemble of similar decision trees, which ultimately vote together on the final synthesis classification. Using hyperparameter optimization, we determined that 20 RF trees give the best model performance. To visualize how our model classifies different types of synthesis procedures, we show in Fig. 3.2a one out of the 20 learned decision trees in our RF model. In Fig. 3.2a, the decision tree starts from the topmost node, and branches into one of two child nodes according to whether certain topic n-grams exist in a paragraph, as defined by the criterion of each node. We highlight a representative branch from Fig. 3.2a in yellow, and show the enlarged branch in Fig. 3.2b. For a paragraph that has topic "cooling-1" after topic "autoclaving" in two consecutive sentences, the decision tree changes its classification of the synthesis method from "none of the above" to the "hydrothermal" category. Because this "hydrothermal"



B

Category	Precision	Recall	F1
Solid-state	96.9±1.0%	93.4±1.5%	95.1±0.6%
Hydrothermal	97.3±1.0%	98.3±1.0%	97.8±0.6%
Sol-gel precursor	95.5±1.4%	87.0±2.1%	91.0±1.1%

Figure 3.1: RF model learning curves and performances. **(a)** Learning curves of the RF model demonstrating F1 score improves with more training data. The red plus and blue cross symbols represent model F1 scores tested on training data sets and test data sets, respectively. The shaded areas denote the standard deviations of the curve. The performance converges to high F1 scores with training data sets as small as a few hundred paragraphs. **(b)** Precision/Recall/F1 scores of the RF model. The model was trained using 5000 training paragraphs and tested using 1000 hold-out paragraphs. We performed 5-fold cross-validation on the training paragraphs to estimate the standard deviation.

node does not have any child nodes, no more decisions will be made and the decision tree predicts the paragraph as having a hydrothermal synthesis procedure.

In many ways, the RF algorithm classifies materials synthesis procedures similarly to how a solid-state chemist would—by looking for patterns of experimental procedures. For example, "shake-and-bake" is a common pattern for solid-state synthesis. If a paragraph is organized as "mix the precursors and then sinter the mixture", then one would likely classify it as solid-state synthesis. This same classification decision can be found in our computer-generated decision trees, where each node contains a pattern of experimental steps (represented by LDA topic results), such as ("[ball-]milling" $\rightarrow$ "sintering") in the third node of Fig. 3.2b. Moreover, our model represents patterns of synthesis as topic pairs, and we can study how words affect the detection of such patterns. As demonstrated in Fig. 3.2b, when a paragraph contains more keywords of topics "(ball-)milling", "(hot-)pelletizing", and "annealing" than keywords of topics "sol formation" and "solution heating", such as "milling", "pressed", and "annealed", chances are that our model predicts solid-state synthesis instead of sol–gel precursor synthesis.

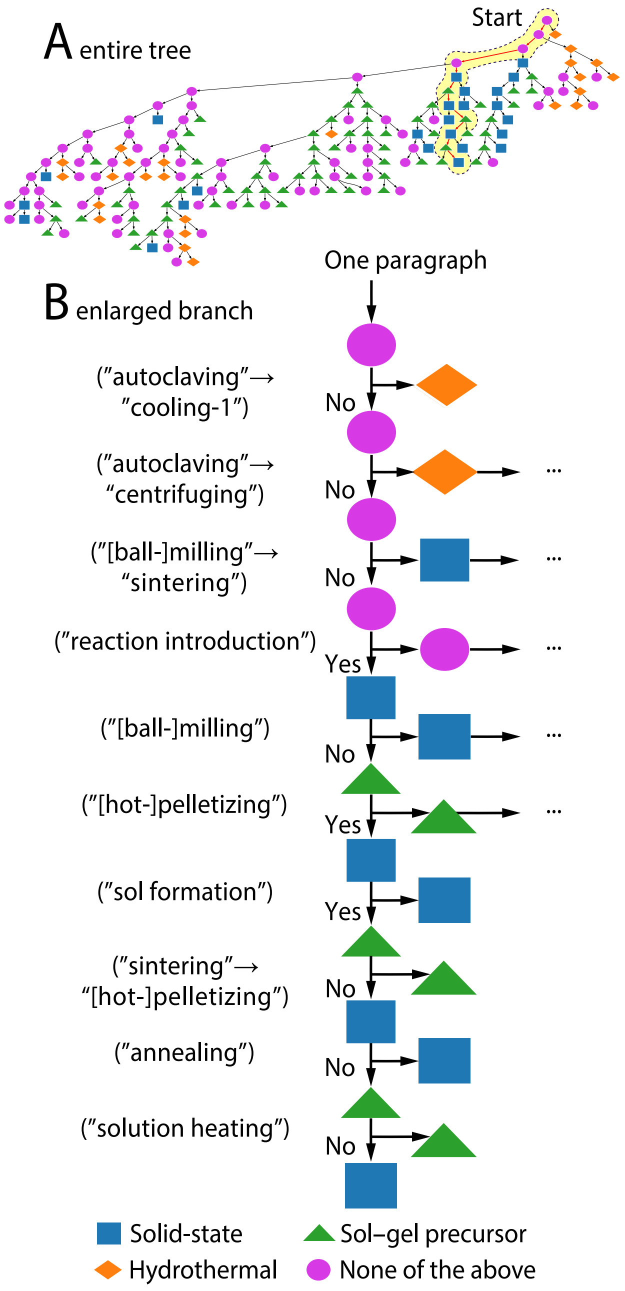


Figure 3.2: Visualization and interpretation of the decision trees in RF. **(a)** The entire decision tree out of 20 trees learned by RF. **(b)** One particular branch. Starting from the topmost node, branch is made when certain topic pairs exist in a paragraph. When no branch can be made, a terminal node predicts the type of synthesis. A RF classifier consists of many trees and selects the majority of predictions.

In general, the features used by the decision trees for classification resemble the underlying procedures of materials synthesis methods. To understand this, we computed the most important features using the built-in feature importance scoring function in *scikit-learn* [157] in Table 3.3. The top 20 important features mostly consist of one-sized and two-sized topic n-grams. As we will discuss in Figure 3.3, these topics and pairs of topics reflect the common nodes and edges in a flowchart representation of synthesis procedures. In this sense, the RF models effectively try to identify the presence of these nodes and edges as classification features, explaining how the RF algorithm automatically picks out features for synthesis procedure classification and weigh them accordingly.

	Feature (topic n-grams)	Importance		Feature (topic n-grams)	Importance
1	autoclaving	0.088	11	pH adjustment	0.032
2	[ball-]milling	0.070	12	[ball-]milling → [hot-]pelletizing	0.027
3	sintering	0.068	13	cooling-1’	0.020
4	autoclaving → washing and dying	0.060	14	sintering → sintering	0.016
5	washing and dying	0.060	15	[ball-]milling → sintering	0.013
6	autoclaving → cooling-1’	0.047	16	[hot-]pelletizing	0.012
7	reaction start → [ball-]milling’	0.045	17	aqueous mixing-2 → sol formation	0.012
8	reaction start	0.044	18	power mixing	0.011
9	aqueous mixing-2	0.040	19	annealing	0.011
10	stirring	0.032	20	sintering → [hot-]pelletizing	0.008

Table 3.3: Top 20 features used in RF as ranked by the feature importance value computed using *scikit-learn*.

However, the above rationale implicitly assume that the synthesis procedures in journal articles are described based on a common representation consisting of ordered experimental steps (as in Figure 3.3). This assumption may not be always valid and, when it’s violated, the RF classifier will have degraded or poor predictive performance. In particular, we have observed several modes of failures by this RF classifier:

- One-sentence description of synthesis procedures. For example, “The target compound was synthesized by using the solid-state reaction of $\mathrm{TiO_2 + BaO \rightarrow BaTiO_3}$.” Since it does not mention any experimental steps, RF classification fails.
- Verbose description of synthesis procedure. If the description of each synthesis step is followed by detailed analysis and arguments, then the topic n-grams will not be able to capture the ordering, since it relies on the adjacency of topics in neighbor sentences but the topics of experimental steps are scattered in different parts of paragraphs in this case.
- Too short sentences. As we will be discussing in Section 3.4, LDA has poor performance when modeling short sentences. If the underlying topic modeling results are

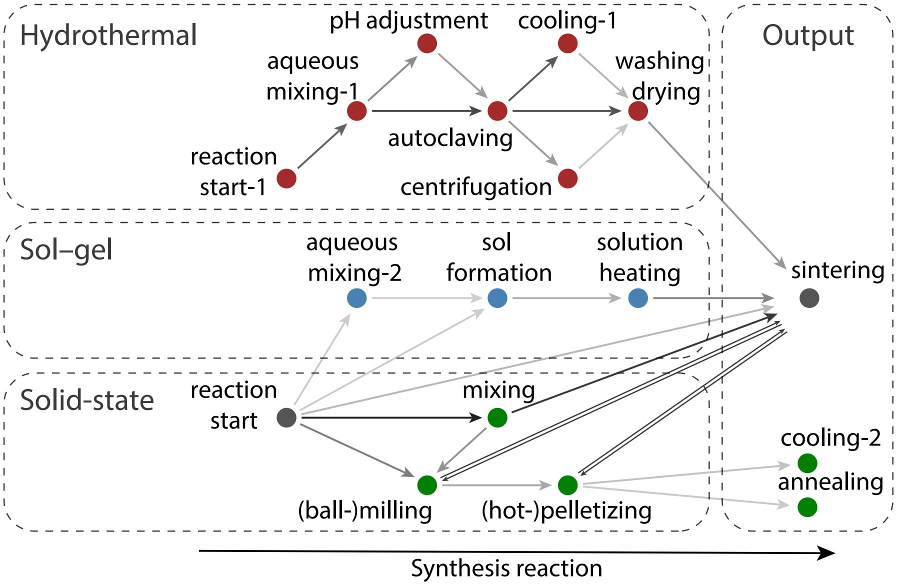


Figure 3.3: Machine-learned flowchart showing the transition between experimental steps for different types of synthesis. The topics associated with the nodes can be found in Table 3.5. Edges represent transitions from one step to another, and the arrows show transition directions. Double-lined edges represent transitions in both directions. A darker edge indicates a more-probable transition.

not accurate enough, then the RF classifier will be fed with unreliable topic n-grams, thus impairing its predictive performance.

3.3 Automated text-mining of synthesis procedure flowcharts

In materials synthesis procedures, experimental steps do not appear randomly—they usually follow a certain procedural order, in patterns that are specific to different types of synthesis methodologies. Similarly, LDA-learned topics do not appear in random sequences in the written synthesis paragraphs. By data-mining the transition probability from one LDA topic to another between adjacent sentences, we can construct a Markov chain representation of how various experimental steps proceed into others. The algorithm used to generate the representation is summarized in Algorithm. 2. We visualize these Markov chains as synthesis flowcharts, shown in Fig. 3.3, using a directed graph consisting of nodes and directed edges, where a node represents an experimental step, and an edge represents a transition from one experimental step to another one.

```
Data: Paragraphs $D^i$ and sentences of each paragraph $s_j^i \in D^i$ annotated with the
      same synthesis type
Data: Topics $T_{j,k}^i$ associated with $s_j^i$ with a probability above threshold $P(T_{j,k}^i) > P_0$
Parameter: $P_0$ Minimum probability of topics to be considered
Parameter: $w_0$ Minimum weight each edge in the final digraph. Edges with
           weights $w < w_0$ are deleted in the final digraph
Result: Digraph $G = (V, W)$ that represents the flowchart of synthesis procedures.
        $V$ is the set of vertices and $W$ is the adjacent matrix
$V \leftarrow \emptyset$;
$W \leftarrow 0$;
/* Build the digraph by filling the adjacent matrix with joint topic
   probabilities.                                                        */
for every paragraph $D^i = D^1, D^2, \cdots$ do
    for every sentence $s_j^i \in D^i$ do
        for every topic $t_j^i = T_{j,1}^i, T_{j,2}^i, \cdots$ do
            for every topic in next sentence $t_{j+1}^i = T_{j+1,1}^i, T_{j+1,2}^i, \cdots$ do
                $V \leftarrow V \cup \{t_j^i, t_{j+1}^i\}$;
                $W[t_j^i, t_{j+1}^i] \leftarrow W[t_j^i, t_{j+1}^i] + P(t_j^i)P(t_{j+1}^i)$;
            end
        end
    end
end
/* Clean up the digraph by removing unimportant edges.                   */
for $t_i \in V$ do
    for $t_j \in V$ do
        if $W[t_i, t_j] < w_0$ then
            $W[t_i, t_j] = 0$;
        else
    end
end
```

Algorithm 2: Pesudocode of the generation of machine-learned flowchart of different type synthesis procedures.

The computer-generated flowchart demonstrated in Fig. 3.3 largely summarizes three types of synthesis procedures. In Fig. 3.3, core experimental steps of syntheses are found, for example, the experimental steps "mixing", "(ball-)milling", "(hot-)pelletizing", and "sintering" (plus "cooling-2" and "annealing") are all found in the solid-state synthesis category, which matches a chemist's intuition of solid-state synthesis. The algorithm also learns important ordering information, for example, "(hot-)pelletizing" usually follows "(ball-)milling", but "(ball-)milling" never follows "(hot-)pelletizing". The edges between "sintering" and "(hot-)pelletizing" or "(ball-)milling" are found in both directions, indicating it is a common practice to regrind and pelletize sintered products in solid-state synthesis. In addition, the algorithm automatically captures subtleties regarding syntheses, for example, that "solution heating" is an intermediate step between "sol formation" and "sintering", which physically is because gel-like precursor states are formed when the particle density in the colloid is increased by evaporating liquid solvent; whereas that "pH adjustment" is an optional step between "aqueous mixing" and "autoclaving", as sometimes, but not always, the formation of the final product depends on specific pH values.

Figure 3.3 reproduces common experimental processes from different synthesis procedures, because LDA allows computers to understand individual experimental steps, and the Markov chain construction enables general procedural orderings to be learned as they were recorded in synthesis paragraphs. However, we note such construction omits several information of materials synthesis that may become critical in practical design of synthesis experiments:

- Each process in Figure 3.3 may have a variety of implementations in different research labs which use different devices. For example, there could be many types of milling, such as hand-milling devices, mechanical milling devices, high-energy ball-milling device, etc. These varieties are not reflected in Figure 3.3 and simply summarized as "(ball-)milling".
- Some infrequently mentioned steps are not displayed in Figure 3.3, such as materials transfer, sieving, casting, etc. The reason that such construction omits the infrequently mentioned steps is because they are less specified in papers, since they do not directly affect the chemistry of the underlying material and authors tend to omit them.

The flowcharts in Figure 3.3 describes the most frequently applied experimental steps of materials synthesis in the literature. However, we note that it is generally more scientifically desired to study those synthesis experiments that do not fit into Figure 3.3. For example, the learned flowchart does not contain the transition "sintering" $\rightarrow$ "cooling", but when such transitions are mentioned, it may suggest the experiment must be carried out under certain controlled cooling rates to avoid impurities. Thus, collecting these "unusual" synthesis experiments and correlating their choice of synthesis steps with synthesis features may reveal important aspects of the underlying synthesis reaction.

3.4 Discussion

Much of the technical content in solid-state chemistry papers is locked-up in the ambiguities of written natural language. Topic modeling algorithms can teach computers to automatically elucidate structure and meaning from these complicated written texts. In this Chapter, we combined unsupervised (LDA) and supervised (RF) machine-learning algorithms to accurately categorize different types of inorganic materials synthesis procedures by topic keywords. LDA can automatically learn keywords associated with specific experimental steps in materials synthesis procedures, which produces topic representations of sentences written in natural language. In this way, we shifted considerable amount of human efforts from identifying and featurizing the experimental steps in English to choosing the topic model hyperparameters and interpreting the topic results. Using these topic representations, we used RF algorithms to classify different synthesis methods with high accuracy, using a relatively modest number of manually annotated synthesis paragraphs. Finally, a Markov chain representation of synthesis processes enables the construction of flowcharts, which capture many of the subtleties involved in inorganic materials synthesis. Because little annotation effort is required, our machine-learning classifier can be readily scaled up to categorize and interpret the millions of solid-state chemistry papers from the scientific literature, which can then be data-mined and analyzed using large-scale informatics tools.

LDA helps achieve high classification performance by reducing the ambiguity of natural language. Oftentimes in English, one meaning can be expressed using different synonyms. This ambiguity of English is also very common in the synthesis literature. For example, "grinding" and "milling" are often used interchangeably in experiment descriptions. LDA is designed to solve the ambiguity problem by identifying the same topic (for example, topic "(ball-)milling" in Table 3.5) in different ways of expression. A major advantage of LDA is that it can learn topic representations without human input. This is in contrast to other NLP methods, such as NER or sentence dependency parsing used in similar works [121, 158], which are supervised classification models that require training on all different synonyms with the same meaning. This training is challenging owing to the limited availability of data sets in materials science with labeled text, meaning there are not enough cases for supervised learning. Another risk of neural networks trained to classify paragraphs is that the large number of parameters could lead to overfitting, and they would be unable to classify paragraphs that use synonyms for synthesis process that were not included in the training set.

One well-known limitation of LDA is that it has poor performance when modeling

Topic name	Cluster of keywords
Annealing	°C, h, min, air, annealed, samples, atmosphere, films, heat, treatment, annealing, furnace, treated, temperatures, temperature
Aqueous mixing-1	g, mL, water, solution, ml, dissolved, added, stirring, distilled, deionized, typical, M, mixed, ethanol, aqueous
Autoclaving	°C, autoclave, h, Teflon, lined, stainless, steel, transferred, microwave, heated, mixture, mL, solution, sealed, min
(ball-)milling	Ball, milling, h, milled, powder, mill, powders, balls, mixed, rpm, planetary, ratio, speed, zirconia, steel
Centrifuging	Water, washed, times, distilled, remove, ethanol, deionized, solution, dried, filtered, centrifugation, precipitate, three, collected, washing
Cooling-2	°C, min, temperature, rate, heating, h, heated, room, samples, cooling, furnace, cooled, K, Cmin-1, sample
(hot-)pelletizing	mm, pressed, diameter, powder, pressure, powders, pellets, pressing, hot, die, thickness, sintered, press, sintering, samples
Mixing	Materials, mixed, purity, starting, powders, stoichiometric, prepared, grade, mortar, amounts, raw, ratio, high, powder, composition
pH adjustment	pH, solution, M, NaOH, adjusted, solutions, buffer, HCl, acid, prepared, aqueous, sodium, phosphate, concentration, added
Reaction start	Prepared, method, solid, state, reaction, synthesized, x, samples, powders, conventional, gel, doped, sol, powder, synthesis
Sintering	°C, h, air, calcined, dried, K, powder, obtained, heated, powders, sintered, finally, samples, furnace, atmosphere
Sol formation	Acid, solution, ratio, added, glycol, water, citric, TEOS, molar, ethylene, prepared, agent, sol, ethanol, titanium
Solution heating	°C, h, mixture, stirred, reaction, heated, temperature, solution, min, stirring, bath, water, room, cooled, oil

Table 3.5: List of topics relevant to solid-state, hydrothermal and sol–gel synthesis procedures. By interpreting the keywords, we assigned a label of experimental steps to each topic. Topics labeled with "*-1/*-2" such as "aqueous mixing-1" and "cooling-2" are merely labeled with the same name but are learned as two independent topics.

topics in short sentences or paragraphs [159]. We observed some incorrect classification results for short paragraphs, but these occurrences are rare, as it is nearly impossible to describe a full synthesis procedure in only a few words, and it is easy to filter all short paragraphs by the length of word sequences.

During the actual deployment of the algorithm, we observed increasing false positives as the articles database was updated to include new papers. This is primarily due to the fact that the original topic model was not trained using the newly added papers. As a result, topic modeling for these new papers had lower quality results, which adversely affected the ML features and classification performance. To solve this, we have a policy in our data extraction pipeline to retrain models when the database makes major revisions.

From the perspective of building an inorganic materials synthesis database, we argued in Chapter 1 that three levels of information are required: high-level classification of synthesis methodologies, intermediate-level experimental steps, and detailed-level processing parameters. We have shown that LDA is well-poised to learn the high-level synthesis methodologies and the intermediate-level experimental steps. However, LDA should be less capable of identifying the detailed-level processing parameters because it is designed to model topics (collections of common objects, ideas, facts [93]), whereas processing parameters appear as single words or phrases and need to be extracted using word-level algorithms, such as NER. Nevertheless, LDA is capable of constraining the problem domain by clustering [160] and smoothing [161] documents, and thus promoting performance of NER tasks [162, 163].

Good examples of mining materials synthesis parameters from journal articles have been previously shown by Kim et al. [63, 158], where they used NER to extract synthesis parameters and applied LDA as a post-processing analysis to cluster the chemistry of materials. These algorithms are trained and evaluated on materials synthesis paragraphs without a specific domain. However, online journal articles describe a large variety of synthesis methodologies, such as the solid-state, hydrothermal and sol–gel precursor syntheses studied in our text-mining project, where different domain knowledge is implicitly assumed, such as the vocabulary of describing experimental steps (Table 3.5) and the organization of these steps (Fig. 3.3). Proper consideration of the subtle domain knowledge is essential for machine learning to understand the synthesis literature in a higher resolution. Our semi-supervised approach allows paragraphs to be automatically clustered into small sub-domains of synthesis methodology, which provides a foundation for codifying domain knowledge and creating a more sophisticated analysis of synthesis information.

3.5 Conclusions

In this Chapter, we demonstrated a semi-supervised machine-learning algorithm for modeling synthesis procedures in journal articles. Our approach benefits from high-classification performance while being trained on data sets small enough to be manually annotated by individual experts. Although this Chapter has focused on a particular case study specifically for classifying materials synthesis paragraphs, the applicability of our method is general. For example, our method can also be used for extracting materials characterization information, which is a valuable text source for identifying the phases of synthesized materials. There are undoubtedly further opportunities to apply topic modeling methods to extract other important data and concepts from scientific articles published in materials science and other fields. We believe that the algorithm presented in this Chapter gives a blueprint for how written information, contained in the large body of published literature, can be extracted and made machine-interpretable.

3.6 Methodology and implementation details

Scientific articles used to develop the algorithms in this Chapter are journal publications published by Springer, Wiley, Elsevier, the Royal Society of Chemistry, and the Electrochemical Society from which we received permissions to download large amounts of articles. For each publisher, we manually identified all materials science related journals available for download. A web scraping engine was built using scrapy (`https://scrapy.org/`). Only full-text articles published after 2000 were downloaded, including metadata such as journal name, article title, article abstract, authors, etc. All data were stored in a document-oriented database implemented using a MongoDB (`https://www.mongodb.com/`) database instance. Because downloaded articles are in HTML/XML format, which contains irrelevant markups and stylesheets, we developed a customized library for parsing article markup strings into text paragraphs while keeping the structures of paper and sections headings. The current snapshot of the database contains 2,284,577 papers [2], from which we used 3,210,525 paragraphs in the experimental sections of each paper to conduct this research. The experimental sections were identified by using case-insensitive keyword matching in section headings. (These keywords are “experiment”, “synthesis”, and their morphological derivations.)

Plain text paragraphs were segmented into sentences and tokenized into words using ChemDataExtractor tokenizer [59], which is purposely trained on scientific corpus to handle abbreviations, chemical formulas, etc. Lemmatization preprocessing [66] was not practiced to keep the meanings of different word forms such as verb fired and noun fire. Common English stop-words serving as grammatical function words such as the, be, on, that were removed from each sentence.

We used the Mallet package [150] to train LDA topic models. Two parameters α and β, which control the Dirichlet prior distribution of the topic distributions and the

[2] The database was expanded a few times. The numbers here only represent the snapshot of the database when this work was performed.

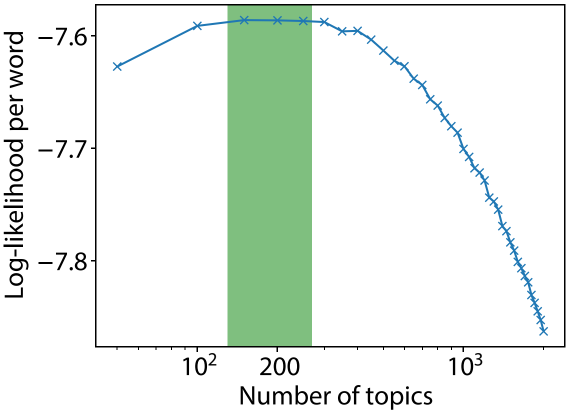


Figure 3.4: LDA model likelihood versus number of topics learned. Using a small number of topics will force LDA to mix different topics, making it difficult to interpret synthesis steps from words distributions; using a large number of topics will cause overfitting by learning many duplicated topics. The best model is obtained by maximizing model likelihood, which suggests using 200 as the number of topics.

words distributions, respectively, were set to $\alpha = 5/N$ and $\beta = 0.01$, where N is the number of topics. Inappropriate settings of the number of topics downgrade the quality of topics learned by LDA. By maximizing LDA model probability likelihood [94], we found that setting the number of topics $N = 200$ produces the best performance of the LDA model without overfitting, as demonstrated by Fig. 3.4.

We used the RF module in the scikit-learn Python package [157] to train classification models. The “topic n-gram” feature is created as indicator variables for n-topic tuples in consecutive sentences $(T_i, T_{i+1}, \cdots, T_{i+n-1})$. Each T_i is a topic in the i-th sentence with probability > 0.05. n denotes the length of the tuple, and we used $1 \leq n \leq 3$ in our study.

The training data set was annotated by synthesis experts in our research group and consists of 1000 training paragraphs for each of the three types of synthesis (solid-state, hydrothermal, and sol–gel precursor synthesis) as well as 3000 randomly sampled negative paragraphs from the database that do not contain any of the above three synthesis procedures. We annotated the data set according to a list of self-consistent definitions developed by us. These definitions can be found in the supplementary material. In total, 6000 annotated paragraphs were obtained. Since annotating scientific papers require extensive training which was time-consuming, we used small groups of annotators and an iterative annotation process. In each round of the annotation process, annotators would mark a paragraph as “uncertain” if he/she feels not confident about the response. Once enough cases were accumulated, annotators gather and resolve annotation ambiguities. By performing this iterative process for multiple times, we have developed a list of self-consistent definitions of the classification labels: solid-state synthesis, sol-gel precursor synthesis, and hydrothermal synthesis. These definitions are summarized as follows:

- Solid-state synthesis: we look for two key experimental steps: 1. Input materials are ground into fine powders using regular mortar or ball-milling devices. 2. Mixture of the fine powders are mixed (usually by pelletizing) and subject to heat treatment.
- Sol-gel precursor synthesis: we look for two key experimental steps: 1. A two-step sol-gel procedure is used to make a precursor state from input materials. (The sol-gel procedure is done by dissolving input materials in liquid to form a concentrated or colloidal solution, which is then dehydrated by evaporation or polymerized using chemicals, in order to form a gel.) 2. The precursor state is then subject to heat treatment, similar to solid-state synthesis.
- Hydrothermal synthesis: we look for two key experimental steps: 1. Input materials are dissolved into liquid solution, which is kept under high-temperature and high-pressure environments (in a vessel such as an autoclave). 2. Solids are collected from the solution and subject to further heat treatment.
- Beside the key steps stated above, additional grinding, pelletizing, and heating may also take place during synthesis to facilitate the formation of target ceramic compounds.

The above definition was used by two persons to create and validate the training dataset of 6000 paragraphs used here.

To generate Fig. 3.3, we obtained sentence topics with probability > 0.05 in our annotated data set of paragraphs, and counted the topic pairs in adjacent sentences, such as "mixing $\rightarrow$ sintering". By collecting all topic pairs, we can compute the probability that one topic pair follows another. This allows us to order a collection of topics into a Markov chain, which can be visualized using a directed graph, where each node is a topic and each edge is a topic pair. We weighted the edges by normalized frequencies of topic pairs observed in paragraphs. Edges with lower occurrence frequencies were plotted with a more transparent stroke in Fig. 3.3, and edges with occurrence frequencies lower than 0.3 were removed from the figure.

3.7 Data and code availability

All codes and data needed to reproduce the results can be found at this repository: `https://github.com/CederGroupHub/synthesis-paragraph-classifier/`. Note that due to publisher licence agreements, the whole paragraphs used for training cannot be distributed publicly. Instead, only the digital object identifier (DOI) strings and the first/last 50 characters are provided to help users identify the corresponding paragraphs.

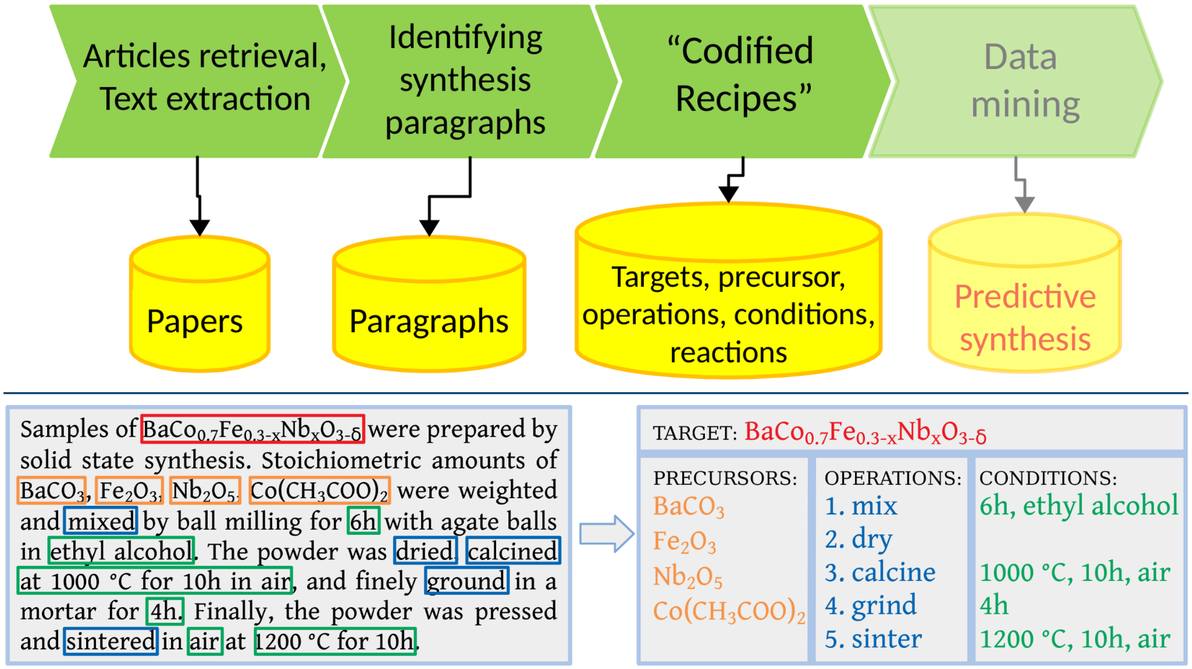
Chapter 4

Compiling a dataset of solid-state synthesis

In this Chapter [1], we describe the text-mining pipeline that was used to generate a dataset of "codified recipes" for solid-state synthesis automatically extracted from scientific publications. The dataset consists of 19,488 synthesis entries retrieved from 53,538 solid-state synthesis paragraphs [2] by using text mining and NLP approaches. The data are collected using an automated extraction pipeline (Fig. 4.1) which converts unstructured scientific paragraphs describing inorganic materials synthesis into so-called "codified recipe" of synthesis. The pipeline utilizes a variety of text mining and NLP approaches to find information about target materials, starting compounds, synthesis steps and conditions in the text, and to process them into chemical equation. Every entry contains information about target material, starting compounds, operations used and their conditions, as well as the balanced chemical equation of the synthesis reaction. The dataset is publicly available in JSON format and can be used for data mining of various aspects of inorganic materials synthesis. Digitizing the large corpus of existing solid-state chemistry literature enables us to make a first step toward development of data-driven approaches for understanding inorganic materials synthesis and synthesizability.

[1] This Chapter is based on the previously published paper by Olga Kononova, Haoyan Huo, Tanjin He, Ziqin Rong, Tiago Botari, Wenhao Sun, Vahe Tshitoyan, and Gerbrand Ceder. "Text-mined dataset of inorganic materials synthesis recipes." *Scientific Data*, Volume 6, Issue 1 (2019) [68] with permission from the authors.

[2] The dataset has been updated a few times and the latest snapshot contains 30,031 chemical reactions retrieved from 95,283 solid-state synthesis paragraphs.


Articles retrieval, Text extraction
Identifying synthesis paragraphs
“Codified Recipes”
Data mining
Papers
Paragraphs
Targets, precursor, operations, conditions, reactions
Predictive synthesis

Samples of $BaCo_{0.7}Fe_{0.3-x}Nb_xO_{3-\delta}$ were prepared by solid state synthesis. Stoichiometric amounts of $BaCO_3$, Fe_2O_3, Nb_2O_5, $Co(CH_3COO)_2$ were weighted and mixed by ball milling for 6h with agate balls in ethyl alcohol. The powder was dried, calcined at 1000 °C for 10h in air, and finely ground in a mortar for 4h. Finally, the powder was pressed and sintered in air at 1200 °C for 10h.

TARGET: $BaCo_{0.7}Fe_{0.3-x}Nb_xO_{3-\delta}$

PRECURSORS:	OPERATIONS:	CONDITIONS:
$BaCO_3$	1. mix	6h, ethyl alcohol
Fe_2O_3	2. dry	
Nb_2O_5	3. calcine	1000 °C, 10h, air
$Co(CH_3COO)_2$	4. grind	4h
	5. sinter	1200 °C, 10h, air

Figure 4.1: Schematic representation of synthesis “recipes” extraction pipeline. *Top panel:* The pipeline starts with retrieval of HTML content from major publishers which is then parsed into a raw text. Next, paragraphs describing synthesis are identified and classified according to synthesis type. Every paragraph is then processed to extract synthesis “recipe”, i.e. materials, operations and conditions. The output is stored in a database for further data mining. *Bottom panel:* Example of processing a synthesis paragraph into a “recipe”. The key component of “recipe”, such as target and starting materials, synthesis steps and their conditions are found and extracted from the paragraph by different text mining algorithms (see later in this Chapter).

4.1 Scraping a large collection of materials synthesis papers

Scientific publications used in this work are journal articles published by Springer, Wiley, Elsevier, the Royal Society of Chemistry, the Electrochemical Society, and the American Chemical Society, from which we received permissions to download large amounts of web-content. For each publisher, we manually identified all materials science related journals available for download. A web-scraping engine was built using the *scrapy* (scrapy.org) toolkit. Since the full-text articles published before 2000’s are mostly in PDF format, which complicates their parsing, we chose to process only papers in HTML/XML format published after the year 2000. The downloaded content includes the text of the article as well as its metadata such as journal name, article title, article abstract, authors, etc. All data was stored in a document-oriented database implemented using a MongoDB (www.mongodb.com) database instance. Because downloaded articles contain irrelevant markups, we developed a customized library for parsing article markup strings into text paragraphs while keeping the structure of paper and section headings.

4.2 Classification of papers and synthesis paragraphs

To find paragraphs on solid-state synthesis, we used a two-step paragraph classification approach described in Chapter 3 which consists of an unsupervised algorithm to cluster common keywords in experimental paragraphs into "topics" and generate a probabilistic topic assignment for each paragraph, followed by a RF classifier trained on annotated paragraphs. The outsome of the RF is a classification of the synthesis methodology in a paragraph as either solid-state synthesis, hydrothermal synthesis, sol-gel precursor synthesis, or "none of the above". The annotation set consisted of 1,000 paragraphs for each label.

4.3 Extraction of synthesis reactions

A typical synthesis procedure in the solid-state chemistry literature contains information about precursor and target materials, synthesis operations and operation conditions. These items comprise a materials synthesis "recipe" and were extracted from a synthesis paragraph as shown in Fig. 4.1. Our extraction pipeline consists of several algorithms which analyze a paragraph and identify information about materials (final products and starting precursors), synthesis steps performed, and conditions associated with those steps. Finally, target and starting materials as well as synthesis conditions are used to balance a chemical equation representing the synthesis reaction. The next sections provide details on each step of the pipeline.

Material Entities Recognition

We created *MatEntityRecognition* [3], a customized materials entity recognition (MER) model to recognize solid-state synthesis reaction entities. To identify starting materials and final products mentioned in a synthesis paragraph, we implemented a bi-directional long-short term memory neural network with a conditional random field layer on top of it (BiLSTM-CRF) [115, 164] which is able to recognize the meaning of a word based on both the word itself and its context. Extraction was performed in two steps each executed by a different neural network: first we identified all materials entities presented in the paragraph; next we replaced each material with a keyword "<MAT>" and classified them as `TARGET`, `PRECURSOR` or `OTHER` material. Each word input for the BiLSTM-CRF was represented as

[3] `https://github.com/CederGroupHub/MatEntityRecognition`

the combination of a word-level embedding and a character-level embedding. The word embedding is restored from a Word2Vec model [98] trained on ∼33,000 solid-state synthesis paragraphs, while the character-level embedding was randomly initialized and then optimized during the training of the BiLSTM-CRF. As an additional feature in the word representation for the second neural network, we also included chemical information about each material, i.e. number of metal/metalloid elements and a flag indicating whether the material contains C, H and O elements only. This assisted in the differentiation of precursors and targets, as they tend to have different number of metal/metalloid elements and are generally not organic compounds in our dataset. We manually annotated 834 solid-state synthesis paragraphs from 750 papers by assigning each word token with the following tags: “material”, “target”, “precursor”, and “outside” (not a material entity). The annotated dataset was randomly split into training/validation/test sets with 500/100/150 papers in each set. The model parameters were iteratively optimized on the training set using early stopping regularization [165] to minimize overfitting, and the model with best performance on the validation set was chosen.

Synthesis operations

We implemented an algorithm which combines neural network and sentence dependency tree analysis to identify key steps of solid-state synthesis given in the paragraph. The neural network was used to classify sentence tokens into 6 categories: `NOT OPERATION`, `MIXING`, `HEATING`, `DRYING`, `SHAPING`, `QUENCHING`, which are the main operations in solid-state synthesis. To create tokens features, we trained a Word2Vec model [98] on ∼20,000 synthesis paragraphs using the Gensim library [166]. For the Word2Vec model training, the sentences of paragraphs were lemmatized, all the quantity tokens were replaced with a keyword <NUM>, and all the chemical formulas were replaced with keyword <CHEM>. The trained Word2Vec model contains 100 dimensions. We also used the SpaCy library [80] to grammatically parse each sentence and obtain linguistic features of token such as token’s part of speech and its dependency to a root token. These linguistic features were then converted into one-hot encoding feature vectors. The word embedding and linguistic feature vectors were concatenated and fed into a multi-layer perceptron (MLP) model with 1 hidden layer of 16 dimensions. The MLP model was trained using cross-entropy loss with 7 output labels. To train the model, we annotated a data set consisting of 100 solid-state synthesis paragraphs (664 sentences) with manually assigned tokens labels. For training, validation and testing, the annotated set was split into a 70/10/20 fraction, respectively. Next, we used the dependency tree to assign `MIXING` operations as a `SOLUTION MIXING` if its lemma belongs to any solvent-based process (e.g ‘disperse’, ‘dilute’, ‘dissolve’, etc) or has a solution environment (e.g. ‘ethanol’, ‘water’, ‘alcohol’, etc.) in its sub-tree. This was differentiated from a `MIXING` operation which consists of grinding or milling in liquid environment, which was assigned the `LIQUID GRINDING` label.

Mixing and heating conditions

For every HEATING operation, we extracted the values or range of values for time, temperature, atmosphere corresponding to the operation, if they are mentioned in the same sentence. We applied a regular expression approach to find the values of temperature and time, and a keyword-search to find atmosphere. For any operation of type MIXING, we extracted corresponding mixing media and type of mixing device, if they are mentioned in the same sentence. For this, we used the list of materials labeled by MER as OTHER materials, as well as keyword-matching, to find potential device or media substances. The extracted attributes were assigned to both the heating and mixing by using dependency sub-tree analysis. Throughout the text, these attributes are referred as “conditions” of synthesis or operations.

Balancing equations

We developed *ReactionCompleter*[4], a code that balances synthesis reactions. Every material entry was processed with a *Material Parser* [5], which converts the string representing the material into a chemical formula and splits it into elements and stoichiometries. Balanced reactions were obtained from parsed precursors and target materials by solving a system of linear equations. Variables of the linear equations represent molar amounts of materials involved in a reaction, and each equation asserts the conservation of a certain chemical element in the reaction. Besides precursor and target materials, we also included a set of “open” compounds (i.e. the compounds that can be released or absorbed during solid-state synthesis, such as O_2, CO_2, N_2, etc.) which were inferred based on the compositions of precursor and target materials. Whenever a target material was synthesized with a “modifier”, i.e. doping, stabilizing, substituting elements, a note is assigned to the reaction: “target <target_name> with additives <element> via <precursor>”. To solve symbolic equations for materials with variable amounts of chemical elements, we used the Gaussian elimination routines in *SymPy*[167].

4.4 The solid-state synthesis dataset

We scraped a total of 4,204,170 papers, which contained 6,218,136 paragraphs in the experimental sections. The experimental sections were identified by using case-insensitive keyword matching in section headings (i.e. “experiment”, “synthesis”, “preparation” and their morphological derivations). Plain text paragraphs were segmented into sentences and tokenized into words using the ChemDataExtractor tokenizer [59]. After classification, 188,198 paragraphs were found to describe inorganic synthesis, such as solid-state, hydrothermal, sol-gel, co-precipitation syntheses, with 53,538 corresponding to solid-state

[4] https://github.com/CederGroupHub/ReactionCompleter

[5] https://github.com/CederGroupHub/MaterialParser

synthesis. These 53,538 paragraphs and their corresponding abstracts were processed to extract materials, operations, conditions and balance chemical equation as described above.

Code availability

The scripts utilized to classify paragraphs and extract recipes as well as to perform the data analysis are home-written codes which are publicly available at the github repository https://github.com/CederGroupHub/text-mined-synthesis_public. The underlying machine-learning libraries used in this project are all open-source: *Tensorflow* (www.tensorflow.org), *Keras* (keras.io), *SpaCy* (spacy.io) [80], *gensim* (radimrehurek.com) [166] and *scikit-learn* (scikit-learn.org) [157] *ChemDataExtractor* (chemdataextractor.org) [59]

The complete dataset of 19,488 solid-state synthesis reactions is provided as a single JSON file, and it is publicly available at github.com/CederGroupHub/text-mined-synthesis_public. Each record corresponds to a single chemical reaction built from a paragraph describing inorganic material synthesis, and is represented as a JSON object in a top-level list. If a paragraph reports synthesis of several materials or a material with variable substituted elements, the corresponding reactions are split into separate data records. Aside from a balanced chemical equation, the metadata for each reaction include: DOI of the paper from which the reaction is extracted and a snippet (50 first and 50 last characters to facilitate its lookup) of the corresponding synthesis paragraph, chemical information about target and precursor materials used in the reaction, operations and conditions for heating and mixing steps to synthesize the target material. The details of the data format are given in Table 4.1.

The chemical equation for the reaction is stored as a string as well as a list of pairs: chemical substance (`material`) and stoichiometric coefficient (`amount`). The reactants and products are listed in the `left_side` and `right_side`, respectively. If in the original paper the target compound was synthesized with variable substituted elements, the element used in the particular reaction is given in `element_substitution`.

The metadata for target and precursors used to construct and balance the chemical equation are represented by a data structure with the following properties:

- `material_string`: string of material as given in the original paragraph before being parsed into chemical composition.
- `material_formula`: chemical formula associated with the material (given originally or constructed empirically by parser).
- `composition`: chemical composition of the material derived from its formula. Aside from single compound materials, we found that a large portion of the materials (pre-

Data description	**Data Key Label**	**Data Type**
DOI of the original paper	`doi`	*string*
Snippet of the raw text	`paragraph_string`	*string*
Chemical equation	`reaction`	Object (*dict*): - `element_substitution`: - `left_side`: *list* of Objects[1] - `right_side`: *list* of Objects[1]
Chemical equation in string format	`reaction_string`	*string*
Target material data	`target`	Object (*dict*): - `material_string`: *string*, - `material_formula`: *string*, - `composition`: *list* of Objects,[2] - `additives`: *list* of *strings* - `elements_vars`: {`var`: *list* of *strings*} - `amounts_vars`: {`var`: *list* of Objects[3]} - `oxygen_deficiency`: *boolean* - `mp_id`: *string*
List of target formulas obtained after variables substitution	`targets_string`	*list* of *strings*
Precursor materials data	`precursors`	*list* of Objects (See `target`)
Sequence of synthesis steps and corresponding conditions	`operations`	*list* of Objects (*dict*): - `token`: *string*, - `type`: *string* - `conditions`: Object - - `heating_temperature`: *list* of Objects[4] - - `heating_time`: *list* of Objects,[4] - - `heating_atmosphere`: *list* of *strings* - - `mixing_device`: *list* of *strings* - - `mixing_media`: *list* of *strings*

[1] {`amount`: *float*, `material`: *string*}

[2] {`formula`: *string*, `elements`: {`element`: amount of element}, `amount`: *string*}

[3] {`max_value`: *float*, `min_value`: *float*, `values`: *list* of *floats*}

[4] {`max_value`: *float*, `min_value`: *float*, `values`: *list* of *floats*, `units`: *string* }

Table 4.1: Format of each data record: description, key label, data type.

dominantly target materials) are composites, mixtures, solid solutions or alloys, written as sequence of ratio-compound pairs. Therefore, a chemical composition entity is represented by a list of dictionaries where each item is associated with a compound found in the materials formula. The ratio of each compound in the material is given in `amount`, its chemical composition (i.e. element and its fraction) is given in `elements`. If a material is one compound, the list has only one item and `amount`=1.0. If a material is hydrate, the water is added into the `composition` list with the `amount` corresponding to the amount of water molecules (if specified).

- `additives`: list of additive elements (i.e. elements used for doping, stabilization, substitution) resolved from material string.
- `elements_vars`: lists all variable elements and their corresponding values found in the materials.
- `amounts_vars`: lists all variable elements ratios and their corresponding values found in the material formula. The values of each variable are given as a structure with `values` listing specific variable's values, and `max_value/min_value` values if range is given in the paragraph.
- `oxygen_deficiency`: yes/no attribute which reflects if material was synthesized with unspecified oxygen stoichiometry.
- `mp_id`: ID of the lowest-energy polymorph entry in Materials Project database (materialsproject.org) if it is presented there.

To facilitate querying of the dataset, the `targets_string` field contains all target material formulas obtained by substituting `amounts_vars` in the `material_formula`.

The sequence of synthesis steps for the reaction (if specified in the paragraph) is listed as a data structure with the following fields: original token from the text (`token`), its type (`type`) as assigned by classification algorithm (see Methods) and conditions used at this step (`conditions`). If the synthesis step has type `HEATING` then temperature, time and atmosphere conditions are provided in the `conditions` attribute. Temperature and time are given as `values` if discrete values are given, or `max_value/min_value` if a range is given. If the synthesis step is of the `MIXING` type then the mixing device and mixing media are specified in the `conditions` attribute.

All the codes and data sets required to train the models are also included in the GitHub repositories. Due to publisher licence agreements, we cannot distribute the entire paragraphs and have provided the DOI of the papers and the first/last 50 characters such that the paragraphs can be identified.

4.5 Technical Validation

Extraction accuracy

The overall extraction yield of the pipeline is 28%, meaning that out of 53,538 solid-state paragraphs, only 15,144 of them produce a balanced chemical reaction. As a test of the full extraction pipeline, we randomly pulled 100 paragraphs from the set of paragraphs classified as solid-state synthesis, and checked them against completeness of the extracted data. Out of the 100 paragraphs, we found 30 that did not contain a complete set of starting materials and final products, meaning that a human expert would not be able to reconstruct a reaction from these paragraphs. The remaining 70 paragraphs could potentially contribute to the dataset as they provide all information about starting materials and final products. Inspections of those 70 paragraphs showed that 42 potential reactions were not reconstructed due to an incomplete or overcomplete set of extracted precursor/target materials, or a failure to parse chemical composition, which makes it impossible to balance the reaction. The former loss originates from the lower re-call of the MER algorithm which we traded in for higher precision, while the parsing problem occurs due to complicated notation used for a materials entity.

Evaluation of the dataset records accuracy was performed by randomly pulling 100 entries and manually checking each extracted field against the original paragraph. The calculated precision, recall and F1-score for every attribute of the data entry is given in Table 4.2. Overall, we achieved a high accuracy in extraction of targets (precision 97%), precursors (F1-score 99%), operations (F1-score 90%) and balancing reactions (precision 95 %). The lower accuracy of the heating conditions (F1-score $<$ 90%) is mostly caused by the cases where the heating step is missed by the operations extraction algorithm. The retrieval of the mixing conditions show relatively poor accuracy with F1-score 65%. This is largely due to misidentification by MER of the device material or media substance used for mixing, as well as because those conditions are often not mentioned in same sentence as the mixing procedure.

This analysis leads us to a conclusion that at the chemistry level (correct precursors, targets, reactions), the accuracy of the dataset is 93%. When including all operations and their conditions, the accuracy of having all recipe items (chemistry, operations and attributes of the operations) extracted and assigned correctly is 51%, which is low due to low performance in extraction the mixing attributes. For many solid-state recipes, specifics of mixing the precursors is of less importance, so this extraction failure is less critical. When considering only correctness of the recipe without conditions for heating and mixing (i.e. chemistry, operations and reactions), the accuracy rises to 64%.

It is worth noting that for this dataset we aimed to achieve higher precision of the data extraction in expense of lower recall (i.e. better miss the data record, rather than

Data attribute	Precision	Recall	F1 score
Materials			
- targets	0.97	/	/
- precursors	0.99	0.99	0.99
Operations	0.86	0.95	0.90
Heating conditions			
- temperature	0.85	0.87	0.86
- time	0.90	0.88	0.89
- atmosphere	0.89	0.86	0.87
Mixing conditions			
- mixing media	0.62	0.66	0.64
- mixing device	0.82	0.55	0.66
Balanced reactions	0.95	/	/

Table 4.2: Performance of data extraction for dataset entries.

Targets	Precursors	Reactions
$LiFePO_4$	TiO_2	$BaCO_3 + TiO_2 = BaTiO_3 + CO_2$
$LiMn_2O_4$	$SrCO_3$	$3CuO + 4TiO_2 + CaCO_3 = CaCu_3Ti_4O_{12} + CO_2$
$BaTiO_3$	$BaCO_3$	$0.5Bi_2O_3 + 0.5Fe_2O_3 = BiFeO_3$
$BiFeO_3$	La_2O_3	$SrCO_3 + TiO_2 = SrTiO_3 + CO_2$
$CaCu_3Ti_4O_{12}$	$CaCO_3$	$2Li_2CO_3 + 5TiO_2 = Li_4Ti_5O_{12} + 2CO_2$
$SrTiO_3$	Bi_2O_3	$TiO_2 + CaCO_3 = CaTiO_3 + CO_2$
$Li_4Ti_5O_{12}$	Fe_2O_3	$Nb_2O_5 + ZnO = ZnNb_2O_6$
$Y_3Al_5O_{12}$	Nb_2O_5	$6Fe_2O_3 + BaCO_3 = BaFe_{12}O_{19} + CO_2$
$CaTiO_3$	Li_2CO_3	$Li_2CO_3 + TiO_2 = Li_2TiO_3 + CO_2$
$LiNi_{0.5}Mn_{1.5}O_4$	Na_2CO_3	$0.5Li_2CO_3 + 0.333Co_3O_4 + 0.083O_2 = LiCoO_2 + 0.5CO_2$

Table 4.3: Ten most common targets, precursors and reactions present in the dataset.

provide the wrong one), therefore the extraction rate is low. Yet, constructing the balanced chemical equation sets up additional constraints on targets and precursors, and helps to reduce potential errors that may have been caused by composition parsing. This results in a skew of the metrics toward higher accuracy for identification of targets and precursors, as compared to operations.

Dataset mining

In order to test the diversity of the entries representing the dataset, we first obtained a list of unique materials (targets and precursors) and reactions. The dataset contains 13,009 unique targets, 1,845 unique precursors and 16,290 unique reactions. The almost 10-fold lower variety of precursors compared to targets can be explained by the fact that in general researchers operate with a set of common well-established precursors. Table 4.3 represents the ten most frequent targets, precursors and reactions in the dataset. The target compounds neatly capture the types of materials most often studied in the last two decades via solid-state synthesis. These are lithium ion battery cathode materials ($LiFePO_4$, $LiMn_2O_4$ and $LiNi_{0.5}Mn_{1.5}O_4$), as well as perovskites for multiferrorics, LEDs and CMOS applications ($BaTiO_3$, $BiFeO_3$, $SrTiO_3$, $Y_3Al_5O_{12}$). It is possible that this “top-ten” materials list is biased by the set of publishers that gave us permission to access their scientific corpus. For example, The American Physical Society was not included and may have brought other compounds to the list.

Next, we evaluate the chemical space covered by the dataset. For each chemical element, we computed the amount of the reactions which include the given element in the target. The results are mapped in Figure 4.2 in the yellow-to-green gradient frame at the top of each element box. The database is dominated by target materials containing Ti, Sr, Ba, La, Fe – $>$ 3,000 reactions include these targets with these elements. This is also reflected in the list of the ten most frequent target materials appearing in the dataset (Table 4.3). The next-most prevalent targets are materials with Li, Ca, Nb, Mn, Bi – 2,000–3,000 reactions with these elements in targets. The least common elements are Au, Pt, Os, Be – $<$ 13 reactions in the dataset contain these elements. The rare and radioactive elements such as francium, radium, technetium or promethium are not presented in the target materials of the dataset.

We also examined the co-occurrence of chemical elements and the most typical counter-ions in precursor materials, and determined the average firing temperature used with each of these precursors. Here, we operationally define the firing temperature as the temperature used during the last heating step in the sequence of synthesis operations. The results are shown in Fig. 4.2 as bar-graphs for each element. The color of the bar correspond to a specific counter-ion. The pure element as precursor is shown in magenta. The length of the bar denotes the average firing temperature.

From this dataset, we can also obtain the distributions of precursor materials used to synthesize each chemical elements in solid-state chemistry. Alkali and transition metal cations are often introduced into a reaction via a variety of precursors, including binary oxides, carbonates, phosphates, nitrides, sulfides, etc. This is possible because these chemical elements are able to bonded with different type of anions. In addition, it is hypothesized that the usage of different precursors can directly adjust the reaction thermodynamic driving force and thus affect the formation of impurities and final products [33]. At the same time, some of the cations in precursor compounds can be found only in the form of oxides or pure elements (e.g. Be, Sc, Hf, Ru, Os, Rh, Pb, Nb, Pt, Au, . . .). Note that these chemical elements are either rare and expensive, or more chemically inert

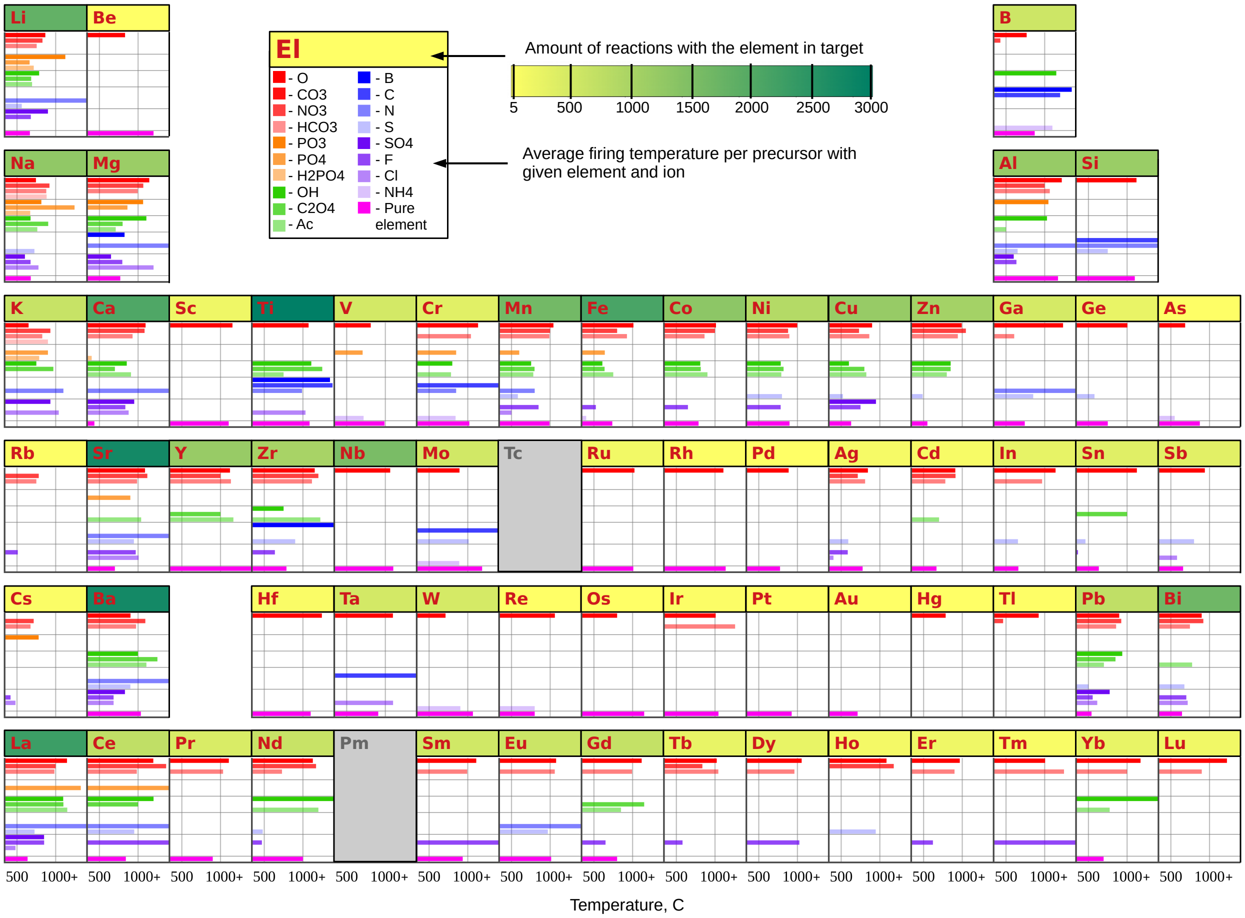


Figure 4.2: Map of chemical space covered by the dataset. For each element, the frame colored in a yellow-to-green gradient represents the total amount of reactions that produce a target compound containing the element. The bar graph below each element shows the list of ions paired with the element in precursor compounds. The length of the bar corresponds to the firing temperature averaged over all the reactions using the given precursor (i.e. element+counter-ion). The elements occurring in five and less targets are faded in grey. “Ac” stands for acetate radical CH_3COO^- in the compound formula.

so precursors with different anions are not possible.

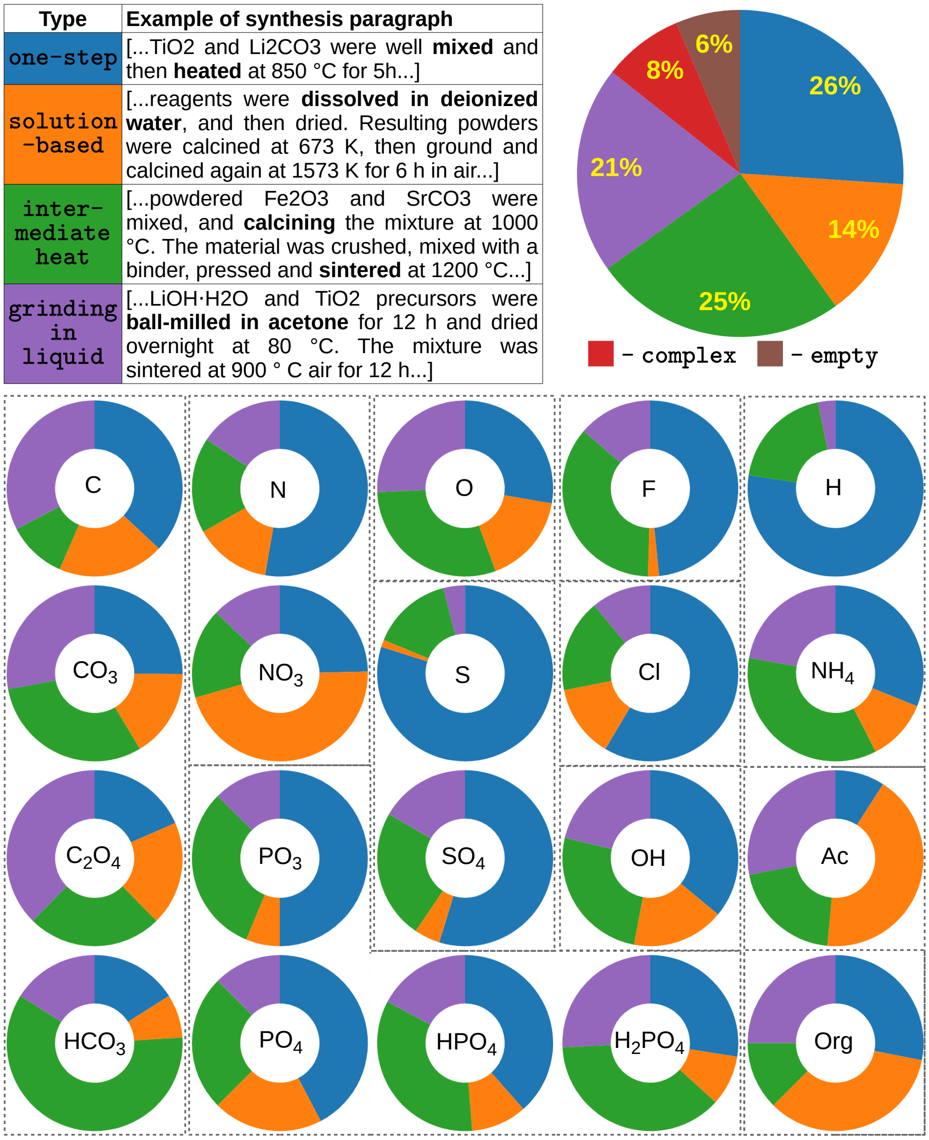
In solid-state synthesis, the counter-ion governs the melting or decomposition temperature of the precursor and may determine when the precursor becomes active during synthesis. The distribution of firing temperatures in Fig. 4.2 agrees well with this statement and illustrates how different precursors are used in different temperature regimes during solid-state synthesis. For example, the blue bars have in general larger length (high average temperature) than red ones, because in general, transition metal borides, carbides and nitrides often have higher reaction temperatures than their corresponding oxides, due to the refractory nature of their precursors, and it's hypothesized that precursor volatility is correlated with the optimal synthesis temperature [33]. We will demonstrate explicitly modeling and prediction of synthesis temperatures in Chapter 5 using features including precursor properties. On the other hand, the green bars are relatively shorter (lower average firing temperature) than red ones, because, compared to oxides and complex oxide anions (carbonates, phosphates, etc), synthesis with hydroxides, oxalates, and acetates facilitate lower temperature reactions as they are often homogeneously mixed by precipitation from solution. This data-driven temperature analysis is based on precursor, and we acknowledge that reaction temperatures also depend on the thermal stability and reactivity of the target compounds. Nonetheless, the figure provides a semi-quantitative starting point for the researchers: If a target material decomposes at relatively low temperature, it may be better to choose a precursor that tends to become active at lower temperature.

In order to demonstrate the diversity of synthesis routes represented in the dataset, we sorted the sequence of synthesis steps according to the following pre-defined patterns (table in Figure 4.3):

- *one-step synthesis* consists of only solid mixing/grinding operations and at most one heating steps (final firing) without regrinding,
- *synthesis with grinding in a liquid media* to homogenize (without dissolution) the starting materials in any liquid media,
- *solution-based synthesis* contains any type of dissolution of starting materials in solvent,
- *synthesis with intermediate heat* has one or more heating steps (not including drying after mixing with liquid part) before final firing of the materials,

First, we found that different synthesis types are represented in the database almost evenly (top pie-chart in Fig. 4.3): 26% of materials are synthesized in one-step, 25% of the syntheses routes are done with intermediate heating step(s) before finial firing, 21% of the syntheses contain grinding (homogenizing) in liquid, and 14% require dissolving of precursors in solvent. The rest of the recipes (14%) either do not contain any detailed synthesis procedure (6%), or the pathway is more complex (8%).

Since the choice of counter-ion used in a precursor often depends strongly on the synthesis method, we surveyed which type of synthesis is common for a specific ion in precursor. We queried a subset of reactions which include the given counter-ion in a precursor compound, and calculated the fraction of each synthesis type in this subset. The resulting pie-charts are shown in Fig. 4.3. The emerging picture is consistent with



Type	Example of synthesis paragraph
one-step	[...TiO2 and Li2CO3 were well **mixed** and then **heated** at 850 °C for 5h...]
solution -based	[...reagents were **dissolved in deionized water**, and then dried. Resulting powders were calcined at 673 K, then ground and calcined again at 1573 K for 6 h in air...]
inter-mediate heat	[...powdered Fe2O3 and SrCO3 were mixed, and **calcining** the mixture at 1000 °C. The material was crushed, mixed with a binder, pressed and **sintered** at 1200 °C...]
grinding in liquid	[...LiOH·H2O and TiO2 precursors were **ball-milled in acetone** for 12 h and dried overnight at 80 °C. The mixture was sintered at 900 ° C air for 12 h...]

Figure 4.3: Correspondence between choice of synthesis route and precursors counter-ions. The top table gives an example of the four synthesis types defined: one-step synthesis, solution-based, synthesis with intermediate heating steps, synthesis including grinding of precursors in liquid media. The pie-charts on the right displays the fraction of each synthesis route in the dataset. The donuts-like charts represent the fractions of the four synthesis routes (given in table) for each counter-ions used in precursors. "Ac" stands for acetate radical CH_3COO^- in the compound formula. "Org" stands for organic radical (-CH-) in the compound formula.

known aspects of solid-state synthesis. For example, in the precipitation of solids during synthesis, the precursor is dissolved in the solution. As shown in Fig. 4.3, the solution-based synthesis (orange fraction) often uses soluble precursors with nitrates, acetates, and organic (CH-containing) radicals. Some counter-ions are more amenable to one-step synthesis than others, for example, chlorides, sulfides, and hydrides do not require much additional processing. On the other hand, relatively stable precursors such as oxides and carbonates are processed in a variety of ways, often requiring intermediate heating and grinding. This is probably due to the common formation of reaction impurities and non-equilibrium intermediates during reaction sequences.

The extraction pipeline we developed allows for automatic processing of scientific paragraphs and identifying key information about solid-state synthesis from there. However, the pipeline still suffers from some issues with the text mining. First, most of the errors down the pipeline are introduced due to incorrect tokenization of the paragraphs and sentences. Although the ChemDataExtractor [59] tokenizer significantly outperforms other NLP packages on chemistry-related texts, it still fails to correctly process large mixtures and solid solutions formulas as well as chemical names consisting of multiple words. We attribute this issue to the fact that ChemDataExtractor was trained on organic chemical entities, and using it for the recognition of inorganic tokens requires modification of the algorithms. Secondly, no established template or pattern exists for describing synthesis procedure which results in significant amount of ambiguity and difficulty when a synthesis method is interpreted even by an expert [168]. This requires development of more advanced text extraction models considering the features of scientific text flow. Third, although the dataset was generated from the paragraphs describing solid-state synthesis (as defined by a classification algorithm), it also contains reactions for solution-based precursors synthesis, such as sol-gel (Fig. 4.3). However, these entries mostly dropped out later in the pipeline, because the majority of them uses organic precursors with complex radicals, and balancing such chemical equations becomes complicated. Lastly, we found that most of the materials studied and synthesized after 2000's are often modified (e.g. doped, elements substituted) compounds, mixtures, glasses or solid solutions. Parsing such materials into composition and building balanced reaction equations is not straightforward. For some compounds with doped and substituted elements, we included the information about modifying elements and corresponding precursors in the reaction string (see Methods). One of the ways to reconstruct reactions for mixtures, solid solutions, alloys, etc. is to split the entire material into compounds and match them with the corresponding precursors. Rather than fully resolve it, we choose to setup a flexible data structure which allows for its further development by the user.

4.6 Conclusion

In this Chapter, we described an IR pipeline to generate ~ 30K codified dataset of solid-state synthesis from ~ 100K synthesis paragraphs (paragraphs in papers that contain synthesis information). This pipeline takes the list of synthesis paragraphs produced in Chapter 3 as inputs, then extracts different attributes of a synthesis experiment into a machine-readable format. The output of this Chapter contains the balanced chemical reaction and detailed properties of precursor and target materials, as well as a list of experimental operations and conditions used to synthesized the target material.

Our pipeline focuses on achieving high precision by sacrificing recall, since we applied several data curation methods such as chemical parsing, chemical formula normalization, and chemical reaction balancing. Tested by human annotation, our dataset achieve >95% precision for balanced reactions and precursor/target materials, and >85% precision for most of experimental operations and attributes. We also automated the entire pipeline such that any new papers can be readily classified by the algorithm in Chapter 3 and extracted by the algorithm in this Chapter.

There are two major applications for the dataset generated in this Chapter:

- First, such datasets can be readily used to set up synthesis search websites. For example, we had created the Synthesis Explorer app on Materials Project `https://next-gen.materialsproject.org/synthesis` that allows users to search our dataset. Researchers can easily find previous papers on certain types of materials that match query criteria in order to learn and design new synthesis experiments, which is not possible with traditional search engines such as Google Scholar.
- Second and the most importantly, machine-readable datasets can be used to data-mine or machine-learn synthesis rules or even build predictive models that automatically propose synthesis experiments [19]. Such efforts are active research directions to help translate the tedious and laborious synthesis knowledge acquisition and lab trial-and-error process using computer automation and machine learning. In the next Chapter, we will discuss a method that trains ML models using the dataset developed here to predicts two important experimental conditions, solid-state synthesis temperature and time. Note that considerable amount of synthesis domain knowledge is required to augment the dataset, as the physical laws that govern the synthesizability cannot be text-mined from the literature.

Chapter 5

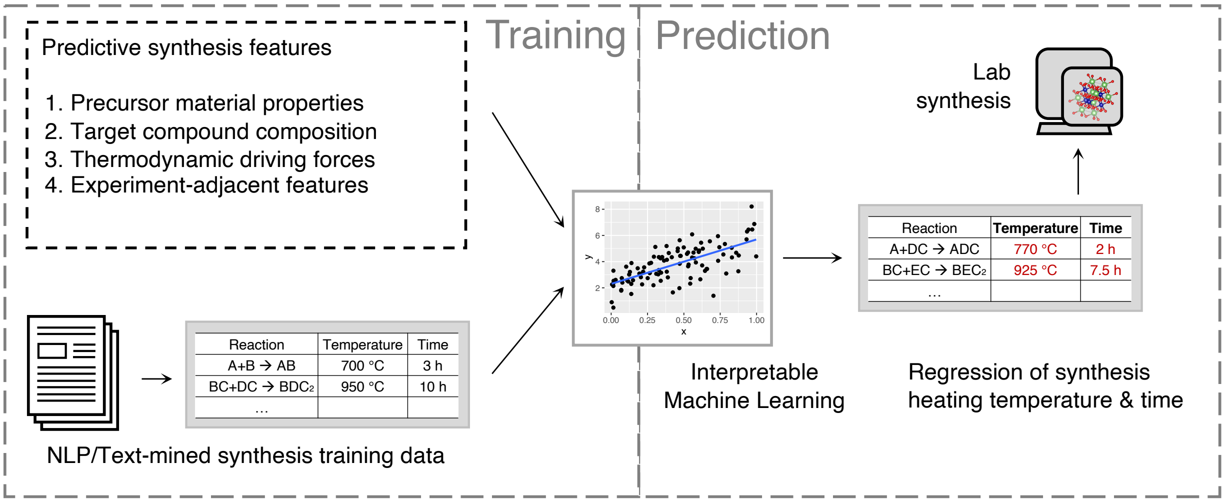
Machine-learning solid-state synthesis condition prediction

5.1 Introduction

There currently exists no efficient methods to determine the appropriate conditions for solid-state synthesis. This not only hinders the experimental realization of novel materials but also complicates the interpretation and understanding of solid-state reaction mechanisms. In this Chapter [1], we use statistical ML methods to systematically learn and quantitatively evaluate synthesis condition predictors from experimental results. Such ML approaches require large, high-quality synthesis datasets covering many chemistries, which have only recently become available through the application of NLP and information retrieval techniques on the large body of scientific literature [63, 68, 100, 109, 158, 169]. In this Chapter, using the dataset of over 30,000 text-mined solid-state synthesis reactions (denoted as the text-mined “recipes" (TMR)) [68], we demonstrate an inductive ML approach that learns synthesis conditions from the knowledge parsed from the past literature.

The overall pipeline of our ML approach is shown in Fig. 5.1. Datasets of synthesis conditions compiled from NLP/text-mined datasets are used to train ML models. Each synthesis reaction was represented using a set of human-designed features, which will be discussed in more detail in subsequent sections. Interpretable ML models were trained on this basis of features to predict two key solid-state synthesis conditions that must be specified for any reaction: heating temperature and heating time.

[1] This Chapter is based on the working paper by Haoyan Huo, Christopher J. Bartel, Tanjin He, Amalie Trewartha, Alexander Dunn, Bin Ouyang, Anubhav Jain, and Gerbrand Ceder. “Machine-learning rationalization and prediction of solid-state synthesis conditions.”, which is pending publication at the time of filing this thesis.


Training
Prediction
Predictive synthesis features
1. Precursor material properties
2. Target compound composition
3. Thermodynamic driving forces
4. Experiment-adjacent features
NLP/Text-mined synthesis training data
Interpretable Machine Learning
Lab synthesis
Regression of synthesis heating temperature & time

Reaction	Temperature	Time
A+B → AB	700 °C	3 h
BC+DC → BDC$_2$	950 °C	10 h
…		

Reaction	Temperature	Time
A+DC → ADC	770 °C	2 h
BC+EC → BEC$_2$	925 °C	7.5 h
…		

Figure 5.1: Schematic of the ML methods developed in this work for predicting solid-state synthesis conditions.

Throughout this Chapter, the prediction of solid-state synthesis conditions is defined as regression (point estimations) of the two experimental condition variables - temperature and time. Several assumptions have been made to simplify the problem: a) Good *synthesizability* is assumed [12, 13, 170, 171], i.e., when a previous publication reports the synthesis of some material at a specified set of conditions, we assume that this reaction was successful. b) Synthesis experiments are performed in a *one-shot* fashion, i.e., reactants react and form the target compound in a single heating step, such that a simple synthesis route of “mix and heat" would be sufficient. c) The ML models predict the “optimal" synthesis conditions as implicitly defined by the consensus of training data.

Note that the above assumptions oversimplify the synthesis condition prediction problem. These assumptions are often violated in many cases of practical solid-state syntheses. For example, a simple one-shot reaction route can thermodynamically favor an impurity phase which can only be avoided by using a multi-step synthesis with specific intermediate compounds [35, 172]; solid-state syntheses are often performed with many more degrees of freedom, such as special heating schedules [32, 172], special mixing devices [173], different sintering aids [21], etc. Moreover, heating atmosphere strongly affects target material formation by changing the chemical potentials of gas species [174]. ML models require sufficient and consistent data to draw statistically significant conclusions [175, 176], while the dataset used in this Chapter has too imbalanced distributions for these additional labels. For example, only $< 5\%$ of the reactions in the TMR dataset have non-air synthesis atmospheres. Therefore, the aforementioned conditions, although are present in the TMR dataset, are not predicted by the ML models in this work. Modeling of these factors will become possible as text-mined datasets become abundant in the future

[64].

In this Chapter, we considered 133 synthesis features describing four aspects of solid-state syntheses: 1) precursor properties, 2) composition of the target material, 3) reaction thermodynamics, and 4) experimental procedure setup. We ranked these features according to their predictive power using dominance importance (DI) analysis [177]. The features were used to train linear and non-linear (tree-based) regressors for synthesis heating temperature and time. We performed leave-one-out cross-validation (LOOCV) to diagnose model performance. We also used out-of-sample (OOS) evaluation on Pearson's Crystal Data (PCD) [178] (another synthesis dataset independently extracted from the literature) to test model generalizability on unseen datasets.

Our ML results achieve a goodness-of-fit measured by $R^2 \sim 0.5 - 0.6$ and mean absolute error (MAE) $\sim 140°\mathrm{C}$ for heating temperature prediction. For heating time prediction, the time variable is transformed into a new prediction variable representing reaction speed: $t \rightarrow \log_{10}(1/t)$. The goodness-of-fit for this new time variable is $R^2 \sim 0.3$ and MAE is $\sim 0.3\log_{10}(h^{-1})$ (e.g., if the predicted time is t, the MAE estimates a range of $[10^{-0.3} \cdot t, 10^{0.3} \cdot t]$, or $[0.5t, 2t]$). Analysis of the model predictive power reveals that heating temperature prediction is dominated by precursor properties, which we hypothesize to be linked to reaction kinetics. Heating time prediction is dominated by experimental operations, which may be indicative of human selection bias. The ML methods developed and applied in this work provide a statistically rigorous approach towards learning robust synthesis predictors from large datasets mined from the scientific literature.

5.2 Ranking features by predictive power

In total, we created 133 features in four categories: 1) precursor properties - 12 features calculated from melting points, standard enthalpy of formation ΔH_f^{300K}, and standard Gibbs free energy of formation ΔG_f^{300K} of precursors; 2) composition of the target material - 74 indicator variables representing the presence (1) or absence (0) of different chemical elements in the target compound; 3) reaction thermodynamics - 33 descriptive features of the driving forces for synthesis-relevant reactions constructed by decomposing synthesis into multi-step phase evolution paths using the previously developed principles [28, 32]; 4) experiment-adjacent features - 14 indicator variables representing whether certain devices, procedures, and/or additives were used in the synthesis procedure. See Methods for a more detailed description of how each of these classes of features were computed.

We first use DI analysis [177] to rank the predictive power of these features. In DI analysis, one constructs many linear models that predict outcomes using subsets of features, called submodels. DI analysis then calculates the incremental effect of a feature f_i on submodels that do not use f_i in three different ways. The average partial dominance importance (APDI) value for f_i is computed as the average increase of model performance, measured by R^2, when f_i is added to any submodel that does not include f_i. In other words, APDI measures the averaged gain of predictive power by including a feature. Individual dominance importance (IDI) values are the R^2 of models trained using only one feature and quantify the predictive power of the features by themselves. Interactional dominance importance (IADI) values are the decrease of model R^2 when a feature is removed from the whole model that uses all features, therefore measuring the gain of predictive power by a feature over all other features. All three DI values are computed for both heating temperature and time prediction models and are shown in Fig. 5.2. We split the dataset into carbonate reactions (reactions with at least one carbonate precursor) and non-carbonate reactions (reactions with no carbonate precursors). This is necessary because these two subsets have dissimilar distributions of reaction thermodynamic driving forces, which must be separated to be modeled in linear regression [179, 180].

We first evaluate the predictive powers of the features by themselves, as demonstrated by the IDI values in Fig. 5.2. For heating temperature prediction, Fig. 5.2 (a) and (b) show that the IDI values of the average precursor melting points are significantly higher than those of other features. Average precursor melting points alone achieve $R^2 \sim 0.2-0.3$ for heating temperature prediction. Other features, such as experimental Gibbs free energy of formation at standard conditions ΔG_f^{300K} and experimental enthalpy of formation at standard conditions ΔH_f^{300K} of precursors, are also highly predictive features as measured by IDI. Note that precursor melting points, ΔG_f^{300K}, and ΔH_f^{300K} are likely to be good proxy variables for precursor reactivity. The next set of predictive features as ranked by IDI are compositional indicator variables (e.g., indicating the presence/absence of Li, Mo, Bi, etc.). These features can be understood as chemistry-specific corrections to heating temperatures. Note that ML models aim to reduce prediction errors for the whole training dataset, which is dominated by the elements that are characteristic of large application fields, such as Li (Li-ion batteries) and Ba (perovskite oxides). It is thus not surprising that these most frequently synthesized chemical systems appear at the top of the list in Fig. 5.2 (a) and (b).

For heating time prediction, Fig. 5.2 (c) and (d) show that the IDI of experiment-adjacent features (e.g., indicators of polycrystal synthesis, phosphors, and usage of ball-milling devices) completely outweigh precursor property features. This suggests that heating time is largely controlled by the desired applications (e.g., the need for dense pellets, small particles, single crystals, etc.) and experimental setups rather than reaction mechanisms. Meanwhile, compositional indicator variables still rank second after the experiment-adjacent features, again acting as chemistry-specific corrections.

The blue bars in Fig. 5.2 are IADI values. IADI values measure the gain of predictive power by a feature over all other features. For heating temperature prediction, Fig. 5.2 (a) and (b) show that IADI values are very small for most features. A low IADI value

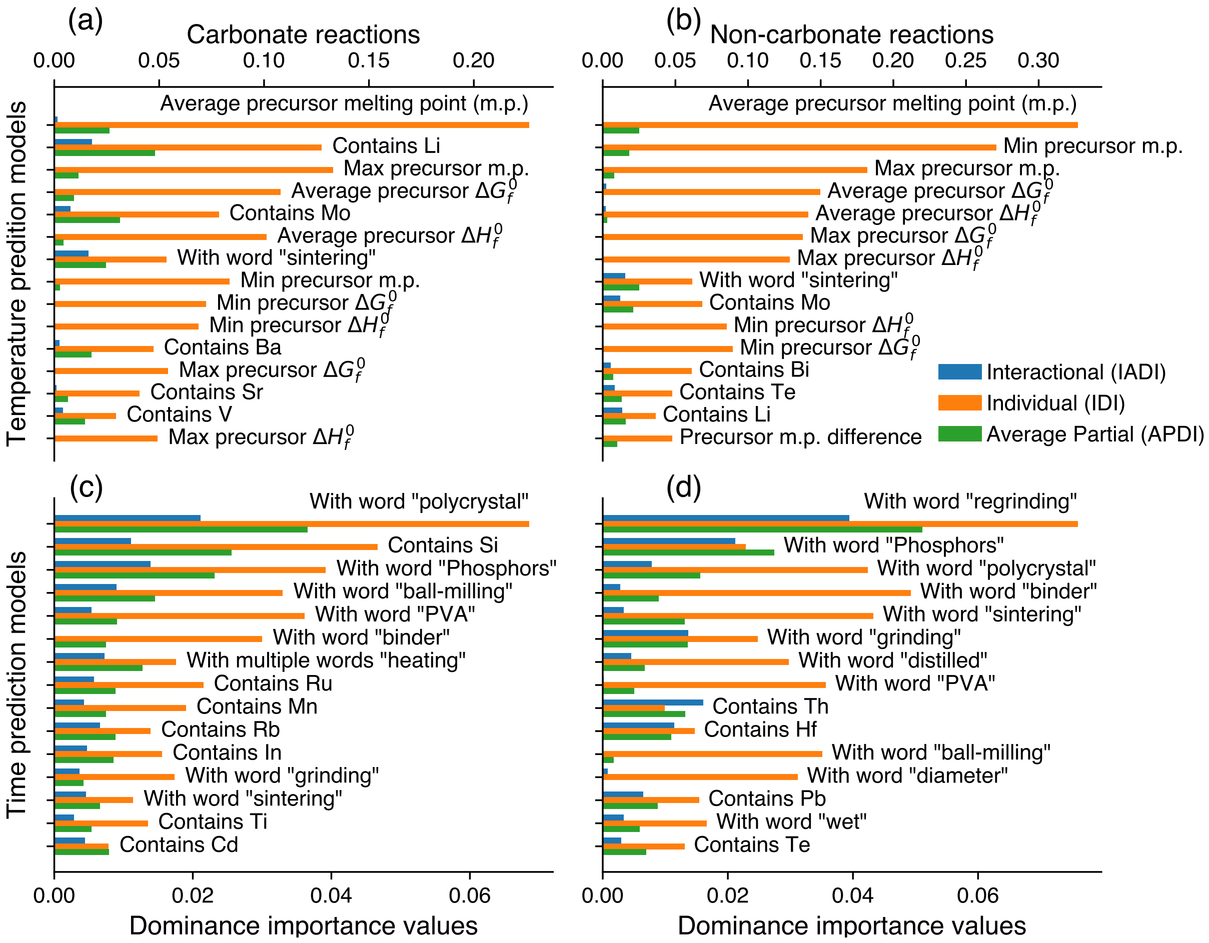


Figure 5.2: DI values and rankings of top 15 synthesis features for heating temperature models (a and b) and heating time models (c and d). The dataset is split into carbonate reactions (reactions with at least one carbonate precursor) (a and c) and non-carbonate reactions (reactions with no carbonate precursors) (b and d). Interactional dominance DI (IADI): decrease of model R^2 when a feature is removed from the whole model that uses all features. Individual dominance DI (IDI): R^2 of models trained using only one feature. Average partial dominance DI (APDI): average R^2 increase when a feature is added to a submodel. Features are ordered according to the sum of all three DI values.

is usually due to high correlation among features, e.g., average precursor melting points and maximal precursor melting points. These high correlations suggest it is necessary to use feature selection to choose the strongest feature among highly correlated features, as will be discussed in the next section. Nevertheless, a few features have relatively higher IADI values, a sign that they bring unique extra information over all other features. For example, describing syntheses using the word “sintering" may suggest the experimenters actively chose higher heating temperatures. As a consequence, the experiment-adjacent feature of “sintering" has the highest IADI value for temperature prediction models.

The green bars in Fig. 5.2 are APDI values. APDI values are the average R^2 increase of a feature to all submodels. Thus, APDI estimates the general usefulness of a feature. APDI and IDI values are therefore two important factors in ranking feature importance. For example, in Fig. 5.2 (a), even though average precursor melting point and ΔG_f^{300K} both have high IDI values, ΔG_f^{300K} has smaller APDI values and is less important due to correlation with alternative features. By ranking all features according to the summation of DI values, we are able to consistently select the most uniquely predictive features.

To summarize, the overall rankings in Fig. 5.2 suggest each prediction variable is dominated by two types of features. For heating temperature prediction, precursor material properties have the most feature importance, while compositional features act as secondary corrections. For heating time prediction, experiment-adjacent features dominate the prediction, while compositional features also provide secondary corrections. Contrary to the common application of decomposing synthesis reactions into multi-step phase evolution paths using thermodynamic principles [32, 34, 35, 37], Fig. 5.2 shows the phase evolution thermodynamic driving force features, developed using similar principles in this work, provide little predictive power for heating temperature and time. We will revisit this result in more detail in Section 5.6.

5.3 Building and interpreting linear regression models

To build regression models, we start with linear regressors as baseline models since their good interpretability allows one to focus on feature engineering and decipher the relations between features and synthesis conditions. To balance between high predictive power and possible overfitting, we add features in the order of DI rankings and drop any feature that increases model Bayesian information criterion (BIC) values [176]. In total, four linear models (heating temperature and time prediction models for carbonate and non-carbonate reactions) were trained using weighted least squares (WLS) [176]. The scatter plots of the predicted synthesis conditions versus the reported conditions are shown in Fig. 5.3 (a) and (b). For heating temperature prediction, the R^2 values of the models are 0.55 on carbonate reactions and 0.56 on non-carbonate reactions, while the MAE are

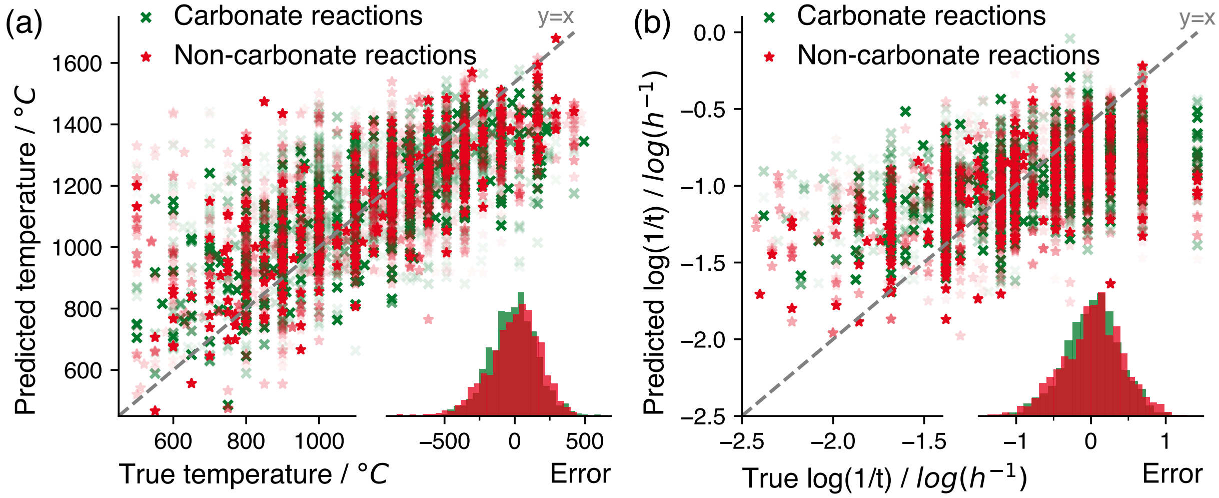


Figure 5.3: Regression result of linear models. The scatter plots show reported conditions v.s. predicted conditions for temperature prediction (a) and time prediction (b). Opacity of the markers indicates the weights of data points. Histograms of prediction errors are also shown.

$134\,^{\circ}\mathrm{C}$ and $147\,^{\circ}\mathrm{C}$, respectively. Using the dataset average (around $1155\,^{\circ}\mathrm{C}$) gives temperature MAE of $198\,^{\circ}\mathrm{C}$ and $220\,^{\circ}\mathrm{C}$, respectively; using the dataset median ($1200\,^{\circ}\mathrm{C}$) gives temperature MAE of $196\,^{\circ}\mathrm{C}$ and $218\,^{\circ}\mathrm{C}$, respectively. For heating time prediction, the R^2 values of the models are 0.31 on carbonate reactions and 0.33 on non-carbonate reactions, while the MAE are $0.30\log_{10}(h^{-1})$ and $0.32\log_{10}(h^{-1})$, respectively. Since we predict the transformed time variable $\log_{10}(1/t)$, such MAE estimates the time prediction is within range $[10^{-0.3}\cdot t, 10^{0.3}\cdot t]$, or $[0.5t, 2t]$ (e.g., for a 2-hour experiment, the expected prediction range is $0.5-4$ hours). Using the dataset average (around $-0.91\log_{10}(h^{-1})$) gives time MAE of $0.36\log_{10}(h^{-1})$ and $0.40\log_{10}(h^{-1})$, respectively; using the dataset median (around $-0.90\log_{10}(h^{-1})$) gives similar results as using the dataset average. Note that these metrics are evaluated on training data. Thus, they may not reflect the model performance when applied on unseen data. We will perform cross validation and discuss the results in later sections.

In a linear regressor $\hat{y}=\sum_i \beta_i x_i$, the feature coefficients β_i quantify how the regression target variable responds to unit changes of x_i. As a special case, when $x_i\in\{0,1\}$ are indicator variables (e.g., compositional and experimental-adjacent features), β_i can be interpreted as additive effects on the prediction target variable when features $x_i=1$. For all compositional features, the effects are shown in Fig. 5.4 (a) and (b). Note that these values are relative to the "average" according to the training dataset and must be interpreted in relative values. For example, if Li is present in the target compound, Fig. 5.4 (a) suggests the heating temperature will decrease by $360\,^{\circ}\mathrm{C}$ on average for non-carbonate

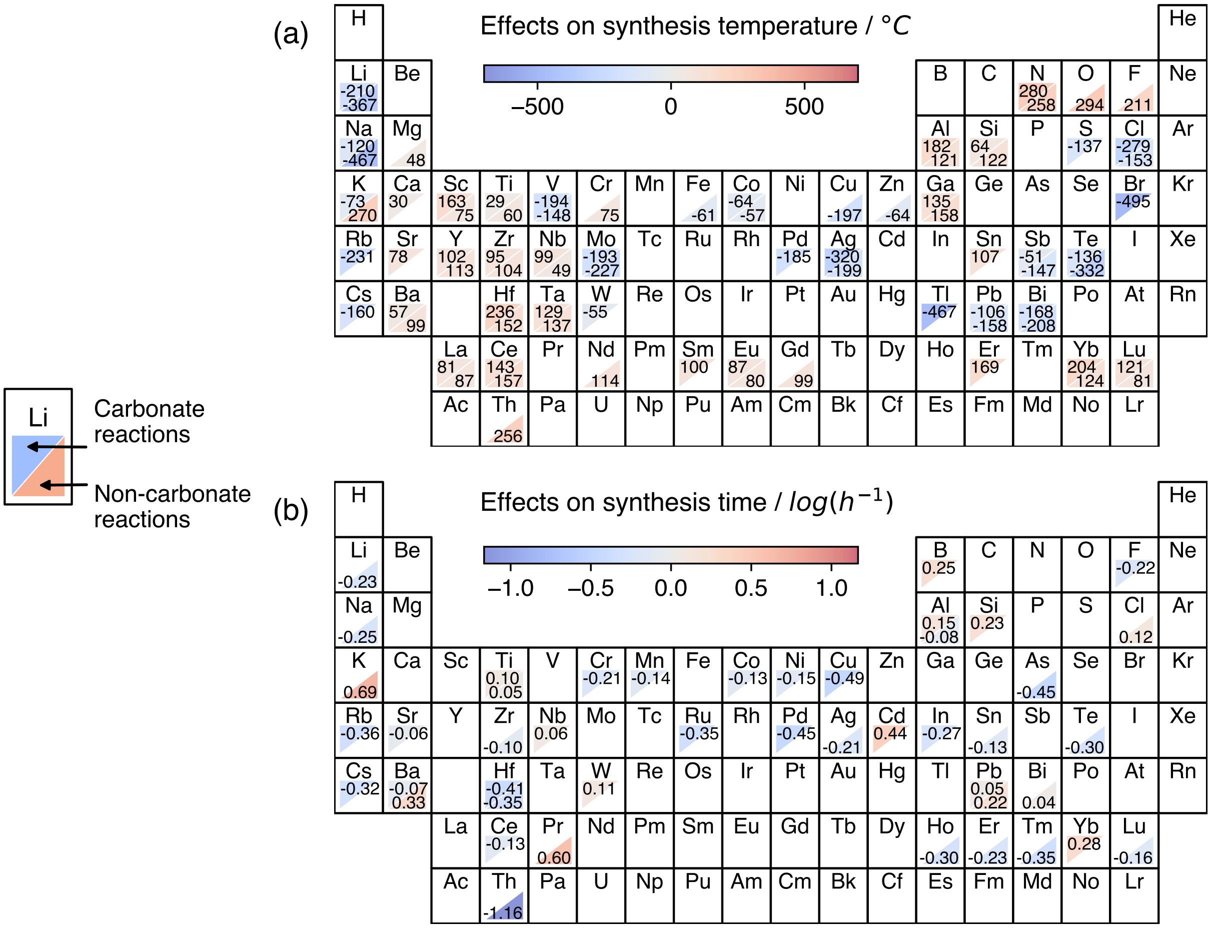


Figure 5.4: The average effect of each chemical element to predicted heating temperatures (a) and times (b) in trained linear models. The values are coefficients of the corresponding features in the linear models, quantifying how much the predicted value changes relatively if a new chemical element is added to (or removed from) the synthesis.

reactions. On the other hand, the presence of N will increase the heating temperature by $260\ ^\circ\mathrm{C}$ on average. Therefore, Fig. 5.4 (a) and (b) are maps that associate different chemistries with their effect on optimal synthesis conditions. Such maps can be used as empirical "synthesis rules" that are helpful for designing synthesis routes to new materials.

The learned coefficients in Fig. 5.4 (a) and (b) are sparse because some elements appear only a few times or are even missing in the training dataset, precluding a confident estimate of their effect (assessed by the p-values of the coefficients with a 5% significance level [181]). In Fig. 5.4, we observe more consistency of compositional effects across similar element periods and groups for temperature predictions than for heating time predictions. The inconsistency of compositional effects for time prediction matches the DI analysis result in Fig. 5.2 (c) and (d), which suggests compositional features are less helpful for predicting heating time. Therefore, compositional features are more likely to capture bias towards particular data points for heating time prediction. Moreover, the compositional effects are less consistent between carbonate reactions and non-carbonate reactions for heating time prediction. These observations suggests the compositional effects are generally less reliable for heating time prediction and must be used with more caution.

5.4 Training and cross-validating non-linear models

Having used DI analysis and linear models to probe the synthesis prediction features, we next aim to systematically cross-validate ML models to understand their generalizability or propensity for overfitting. Fig. 5.5 shows the model performances versus the number of features, which characterize training R^2 and the LOOCV Pseudo-R^2 (a metric comparable to R^2, see Section 5.8) scores of the linear models as more features are included in training. In Fig. 5.5, features are added into the models in the order of DI value rankings. Fig. 5.5 shows that both training and LOOCV scores increase quickly when the number of features is less than 10. This result is consistent with the DI values in Fig. 5.2 as the first few features have the highest feature importance. The model performance continues to improve as we include all other features, although the marginal improvement decreases rapidly. The training and LOOCV curves for linear models exhibit very similar performance, suggesting that these linear models have little risk of overfitting.

The linear model may be incapable of capturing non-linear correlations among features and synthesis conditions. We next use advanced ML models that are capable of modeling non-linear relations on the same set of features as for the linear models. Among many ML models we attempted during preliminary experiments, gradient boosted regression trees (GBRT), implemented in the XGBoost package [182], demonstrated the best LOOCV scores after proper hyperparameter tuning. GBRT models use a large number of weak tree learners to iteratively build a strong ensemble regressor. The model is iteratively

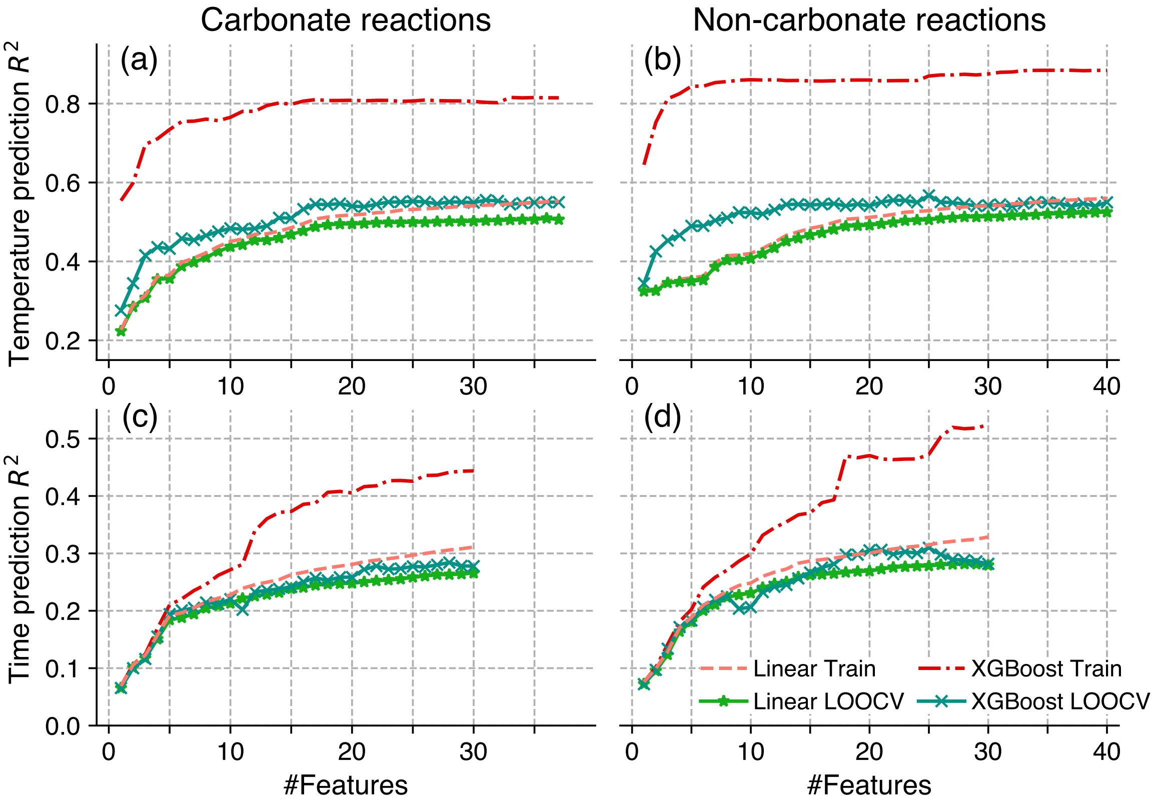


Figure 5.5: Model performance versus number of training features for both linear and non-linear (gradient boosting tree regressor) models. The x-axis shows the number of features used. The features are added in the order of DI value rankings. The first row shows performances of temperature prediction models trained on carbonate reactions (a) and non-carbonate reactions (b). The second row shows performances of time prediction models trained on reactions with (c) and without (d) carbonate precursors.

trained to predict the *gradients*, or the prediction errors, by putting more and more weights on samples that have large prediction errors in a previous boosting step [182]. Compared to the RF algorithm used in Chapter 3, GBRT can better handle outliers and imbalanced datasets, and is generally observed to have similar or better performance for regression problems. We observe in Fig. 5.5 that XGBoost training Pseudo-R^2 (red dashed curves) are significantly higher than linear models. However, as shown by the teal crosses in Fig. 5.5, compared to the LOOCV scores of linear models (green stars), the LOOCV Pseudo-R^2 scores of XGBoost models do not improve as much when compared to the LOOCV performance of the linear models, suggesting an increased level of overfitting by XGBoost models. One advantage of XGBoost over linear models is improved utilization of a small number of features, as shown by the steeper curves when the number of features is less than 10 in Fig. 5.5 (a) and (b), although the advantage diminishes once sufficiently many features are used.

5.5 Testing model generalizability using the PCD dataset

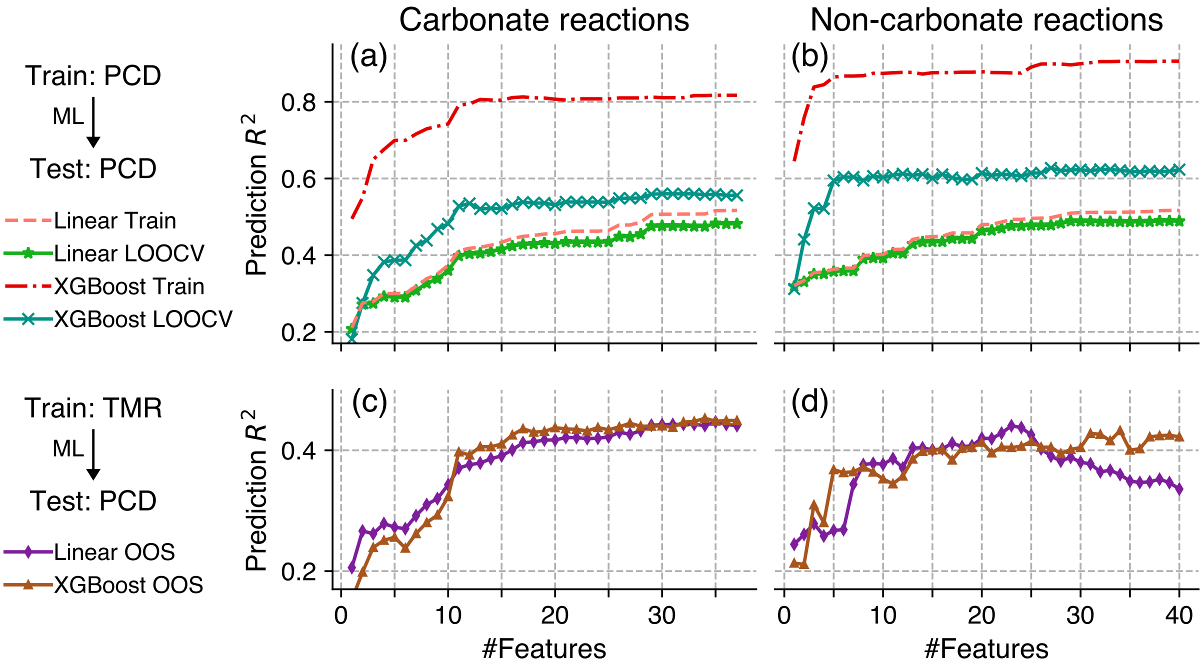


Figure 5.6: Performance of the models versus the number of features evaluated on the PCD dataset. X-axes show the number of features used in each model. Features are added in the order of DI value rankings as in Fig. 5.2. The left panels (a) and (c) show models trained on carbonate reactions and the right panels (b) and (d) show models trained on non-carbonate reactions. Top panels (a) and (b) show performance of models trained and evaluated on the PCD dataset, which represent the upper bounds of OOS scores (c) and (d), which show performance of the models trained on the TMR dataset. A higher OOS score indicate better model generalizability.

When applied to unseen datasets, ML model predictions tend to have larger errors due to dataset shift, i.e., unseen datasets have a different distribution than the training datasets [183]. In particular, the relations between features and outcomes may change for unseen data, leading to *concept drift*, degrading model generalizability and limiting model applicability.

The TMR dataset mostly contains syntheses for inorganic oxide materials and is dominated by target materials containing Ti, Sr, Li, Ba, La, Nb, Fe, etc., reflecting popular materials in the inorganic materials research community such as perovskite oxides and battery materials. The TMR dataset also contains a large fraction of solid solutions or doped materials. To estimate and understand how the ML model trained on the TMR dataset generalizes to unseen datasets, we utilized the PCD dataset as an additional test. The original PCD collection contains inorganic materials syntheses that were manually extracted from the literature in a semi-structured natural language form [178]. We processed the PCD data collection using the same text-mining pipeline and only kept oxide syntheses such that the final PCD dataset has a similar chemistry distribution as the TMR dataset. To ensure there are no duplicate syntheses, we removed any entry in the PCD dataset whose DOI is present in the TMR dataset (i.e., syntheses in same papers are not allowed, but the same compositions from different papers are allowed). Compared to the TMR dataset, the PCD dataset shares a similar distribution of chemical systems and synthesis conditions, as indicated by similar sets of popular chemical elements (i.e. Ti, Fe, Sr, Ba, Si, etc.) and average synthesis temperatures around 1200 °C, see Fig. 5.7. The PCD dataset thus represents a reasonable benchmark dataset for our ML models. However, since many reactions in the PCD dataset do not have heating times extracted, we only predicted heating temperatures for the PCD dataset.

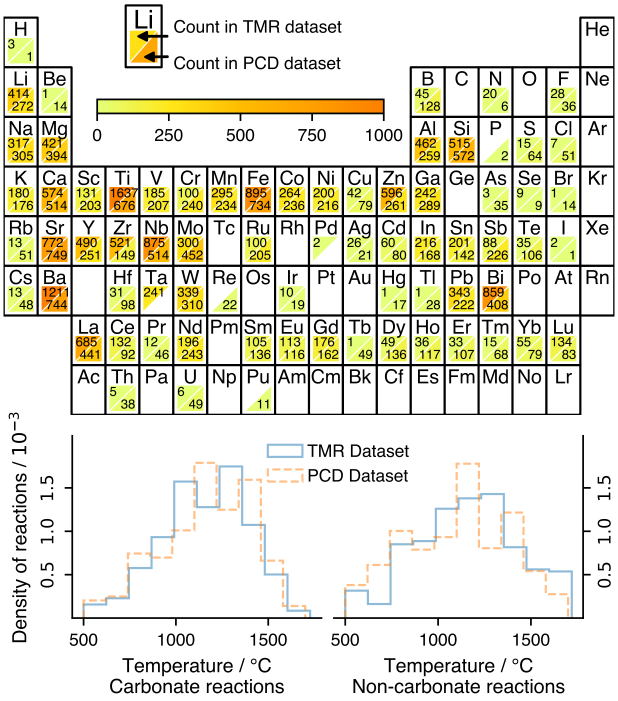


Figure 5.7: Distribution of chemical systems for (Left) the PCD dataset and (Right) the text-mined recipes dataset. O and C are excluded since they appear in all reactions. The bottom panel shows the difference of synthesis temperatures distributions for TMR and PCD dataset.

As a first test, we performed the same training/validation procedure using the PCD dataset to establish an upper bound of the model performance. Fig. 5.6 shows the performance of the ML models versus the number of features. The green stars and teal crosses in Fig. 5.6 are the LOOCV scores of linear and XGBoost models, respectively. XGBoost models achieve $0.5 \sim 0.6$ LOOCV Pseudo-R^2 which is considerably better than linear models ($0.4 \sim 0.5$). Moreover, XGBoost shows steeper performance increase when few synthesis features are used. Compared to Fig. 5.5, the advantage of the non-linear models are much more substantial for the PCD dataset than for the TMR dataset. This clear advantage of XGBoost models indicates they are more robust than linear models against possible dataset shift effects.

Next, we performed tests to understand how well the ML models trained on the TMR dataset are generalizable to the PCD dataset. The purple diamonds and yellow-brown triangles in Fig. 5.6 show the OOS performances of the linear and XGBoost models trained using the TMR dataset but evaluated on the PCD dataset. It is interesting to note that XGBoost and linear models have very similar OOS scores for carbonate reactions, but XGBoost clearly outperforms linear models for non-carbonate reactions when more (> 30) features are used. Upon further investigation, the features #30 to #40 used on non-carbonate reactions are mostly related to thermodynamic properties of the reactions. The performance drop after features #30 suggests that relations between thermodynamic features and heating temperatures learned on the TMR dataset by linear models do not transfer well to the PCD dataset. On the other hand, XGBoost models seem to be able to consistently maintain good performance regardless of the number of features used.

In Fig. 5.6, the difference between LOOCV scores and OOS scores confirms the ML models have degraded prediction performance (R^2 drops by 0.1) when applied to a different dataset. The performance degradation caused by dataset shift is often inevitable and requires regularly retraining the ML models in order to adapt to the new datasets. However, Fig. 5.6 suggests XGBoost models are more robust against dataset shift and have a better generalizability. We hypothesize this is due to the strong regularization and therefore recommend ML synthesis condition predictors to be built with XGBoost or similarly regularized models.

5.6 Discussion

ML predictions must be statistically evaluated using large datasets, so this work has focused heavily on reducing the expected prediction errors and improving the coefficient of determination R^2. We do not optimize models for any particular reaction but aim at predicting the synthesis conditions over a dataset of several thousand synthesis reactions. As demonstrated by the cross-validation and OOS evaluations in Fig. 5.5 and Fig. 5.6, our models achieve $R^2 \sim 0.5 - 0.6$ (MAE $\sim 140\,^{\circ}\mathrm{C}$) for heating temperature predictions and

$R^2 \sim 0.3$ (MAE $\sim 0.3 \log_{10}(h^{-1})$) for heating time predictions.

Based on the ranking of DI values in Fig. 5.2, the deciding factors for the synthesis conditions can be organized into a two-level hierarchy. Synthesis temperature prediction is dominated by precursor properties, which we speculate are proxies for reactivity stemming from the mobility of ions, with additional corrections learned for different chemistries. Synthesis time prediction is dominated by experiment-adjacent features that are linked to experimental setups/intentions, also with corrections according to chemistry. The features used in this work to account for reaction thermodynamics were inspired by recent efforts to understand phase evolution during synthesis [28, 32, 33, 37, 184]. These features involve decomposing overall synthesis reactions into a sequence of phase evolution reactions between pairs of compounds and quantifying the grand potential thermodynamic driving force for these phase evolution reactions. This approach has been proved especially useful for understanding phase evolution pathways observed in *in-situ* experiments. However, in this work, they are shown to provide little predictive power of synthesis conditions and even cause the models to generalize poorly on OOS datasets (as demonstrated in Fig. 5.6). This discrepancy will be discussed in more detail in the subsequent sections.

Connection to Tamman's rule

Our finding that the average precursor melting points are the most predictive feature for heating temperatures is reminiscent of Tamman's rule [185, 186]. Tamman's rule can be formulated as predicting that the synthesis temperature of metal alloys should be more than $\frac{1}{3}$ (for example, $\frac{1}{2} - \frac{2}{3}$) of precursor melting points. This rule is derived from the observation that atomic diffusion quickly ceases below $\frac{1}{3}$ of melting temperatures [2]. Tamman's empirical rule was never formally defined. It is also questionable whether the rule is applicable to the synthesis of ionic compounds in addition to intermetallics. Nevertheless, variants of Tamman's rule are still used to help determine solid-state synthesis conditions. For example, Becker & Dronskowski used $\frac{2}{3}$ of the most “volatile" compound [187] ; other values, such as $\frac{1}{2}$, have also been used [186].

Our ML framework allows us to formally model and test Tamman's rule within a statistical approach. We start with Tamman's original formulation and fit a linear model without an intercept term:

$$T_{\text{Tamman}} = \alpha(\min T_{\text{melt}}) + \varepsilon,$$

where T_{Tamman} is the predicted heating temperature, $(\min T_{\text{melt}})$ is the minimum of precursor melting points, α is a parameter to be learned, and ε is an error term. Both the pre-

[2] The original German text by Tamman is “Die Zahl der Platzwechsel in der Zeiteinheit nimmt vom Schmelzpunkt an mit sinkender Temperatur schnell ab und wird bei Metallen bei Metallen bei 1/3 der absoluten Schmelztemperatur unmerklich." which translates to “The number of changes of place in the unit of time decreases rapidly from the melting point with falling temperature and becomes imperceptible for metals at 1/3 of the absolute melting temperature."

diction and the melting points are presented in degrees Kelvin. The fit linear model finds $\alpha = 1.2$ when trained on carbonate reactions and $\alpha = 0.8$ when trained on non-carbonate reactions. These α values are larger than the commonly used values for Tamman's rule, such as $1/2$ and $2/3$, suggesting the required temperatures for atoms to diffuse significantly in ionic compounds are higher than in intermetallics, or that for ionic compounds, Tamman's rule is a surrogate for another property than diffusion. On the other hand, fitting a linear model with intercept added, $T_{\text{Tamman}} = \alpha(\min T_{\text{melt}}) + \beta + \varepsilon$, finds $\alpha = 0.2$ and $\beta = 1200\text{K}$. The α values in models with an intercept is much smaller than those without an intercept, because the intercept has shifted the temperatures for all syntheses.

The above linear models are not the model with highest predictive power (R^2 values), and it's also unclear if one should use average melting points or minimal melting points, and whether an intercept should be added. Using the previously described DI analysis, we were able to resolve these questions by identifying the model with highest R^2 value. As shown in Fig. 5.2, using average precursors melting points (instead of minimum precursor melting points) yields the highest prediction performance. Therefore, we update Tamman's rule to give the optimal synthesis temperature T_{Tamman} as proportional to the average of precursor melting points $(\text{avg}\, T_{\text{melt}})$ plus a constant. Mathematically, the predictor is defined as:

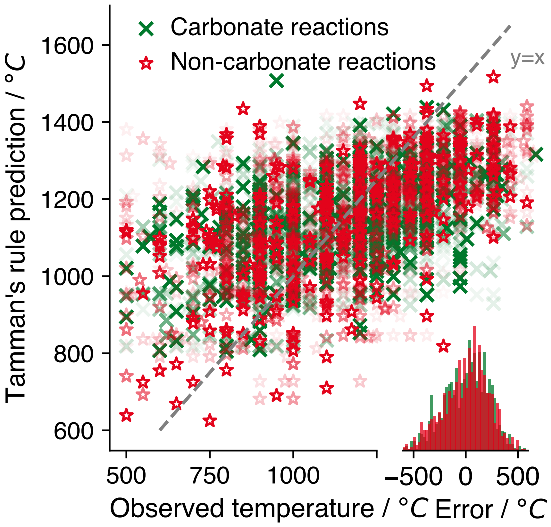
$$T_{\text{Tamman}} = \alpha\,(\text{avg}\, T_{\text{melt}}) + \beta + \varepsilon,$$

where α, β are parameters to be learned and ε is an error term.

As demonstrated in Fig. 5.8, fitting a linear model reveals a slope of $\sim 1/3$. Since we used the average of precursor melting points, the predicted heating temperatures should be generally larger than $\frac{1}{3}$ of the minimal precursor melting point. The predicted versus reported heating temperatures and the histogram of prediction errors are shown in Fig. 5.8 (a). The parameters of the fitted linear model are shown in Fig. 5.8 (b). The large F-statistic values and very small p-values show strong statistical significance of the model although this is contrasted by the low coefficient of determination ($R^2 \sim 0.2 - 0.3$). Note that the formula fitted here should be differentiated from Tamman's original observation [185] since we added intercept and used Celsius temperature scale. Therefore, we name our formula as "extended Tamman's rule". The key message by the extended Tamman's rule is the strong positive correlation with precursor melting points. The extended Tamman's rule is not a perfect predictor and has larger prediction errors at low temperatures. However, it contributes more than $\frac{1}{3}$ of the maximal predictive power developed in this work.

Roles of phase evolution reaction analysis in synthesis condition prediction

Predicting heating temperature is of major scientific interest. In solid-state synthesis, the final products are more sensitive to the heating temperature than time, since insufficiently



(a)

Fit formula: $T_{\text{Tamman}} = \alpha \text{ avg } T_{\text{melt}} + \beta + \varepsilon$		
Data subset	Carbonate	Non-carbonate
$\hat{\beta}$	**669.3** °C	**569.9** °C
$\hat{\alpha}$	**0.346**	**0.329**
R^2	0.226	0.327
#Obs	3182	3143
F-statistic	931.7	1526.0
P-value	10^{-179}	10^{-272}

(b)

Figure 5.8: Fitting result of Tamman's rule, i.e., synthesis temperature is proportional to average precursor melting points. **(a)** Scatter plot of the reported v.s. predicted synthesis temperatures and histogram of prediction error. Opacity indicates data point weights. **(b)** Regression parameters and F-test for model significance. A very small p-value indicates that it is extremely unlikely the result is due to random noise.

low or high temperatures lead to incomplete reactions, impurities, or the complete absence of a desired target phase. Thus, heating temperatures are more carefully optimized than heating times, which are often chosen for convenience (e.g., to run overnight). There have been many successful examples where solid-state synthesis pathways are rationalized using the thermodynamics of reactions occurring during heating. For example, thermodynamic driving forces have been used to understand and control phase evolution pathways in Y−Mn−O oxides [37, 184], Y−Ba−Cu−O superconductors [32], Na−Co−O layered oxides [28], and $MgCr_2S_4$ thiospinel compounds [33]. Inspired by these works, we computed features as numerical transformations of the thermodynamic driving forces obtained by decomposing synthesis into multi-step phase evolution paths. Contrary to the success in reconciling experimental observations in these systems, these features are shown to provide no observable predictive powers for general synthesis condition predictions in this work (as shown in Fig. 5.2 and Fig. 5.6).

A low contribution of predictive power does not necessarily negate the effectiveness of phase evolution reaction analysis for understanding solid-state synthesis. It simply suggests that the features developed in this work are not correlated with the synthesis time and temperature over the diverse datasets evaluated in this work. We hypothesize this arises for a few reasons. First, the scale of reaction driving force may dictate the decision boundary of synthesizable/non-synthesizable conditions (e.g., synthesis should not occur at temperatures where the target phase is unstable with respect to decomposition). However, the dataset used here only contains positive experimental results, so the thermodynamic stability of the target under the chosen synthesis conditions is likely already achieved for all data points. Indeed, in the rationalization of *in-situ* synthesis, characterization has been used more to explain the phases observed along the reaction path rather than the specific conditions [28, 32, 184]. Second, once we are in the region of synthesizable conditions, reaction driving force might become insufficient in determining synthesis conditions that lead to "*fast*" reactions. Since a typical lab synthesis needs to be completed in a reasonable period of time, experimenters may decide to raise heating temperatures to facilitate better reaction rates. Indeed, if we calculate the temperature $T_{equilibrium}$ at which the reaction driving force is zero for the overall synthesis reaction (using the grand potential, $\Delta\Phi_{rxn} = 0$) for all the reactions, we found that this theoretical lower bound of heating temperatures $T_{equilibrium}$ is generally much lower than reported experimental T_{exp}. This suggests experimenters actively use $T_{exp} \gg T_{equilibrium}$ to achieve better kinetics. Unfortunately, reaction driving force analysis do not directly provide kinetic information, which is also chemistry-specific. On the other hand, precursor melting points and formation energies (ΔG_f^{300K}, ΔH_f^{300K}) may be correlated to ion transport kinetics as they are indicative of the relative strength of bonds in the solid precursors. This may explain why precursor material properties are the top predictive features for heating temperatures.

Previously, we demonstrated that precursor melting points (akin to Tamman's rule) provide the most predictive power for heating temperatures if only one feature is allowed (see IDI values in Fig. 5.2). We note here that the effectiveness of Tamman's rule may also be due to the aforementioned selection bias [188] towards fast solid-state syntheses (as well as community knowledge of Tamman's rule). This selection bias is inherent in the synthesis dataset used in this work as the literature only reports "fast" and successful solid-state reactions. We note that some recent investigations of solid-state synthesis mechanisms [32, 189] have put more emphasis on modeling reaction speeds. In addition, with the recent developments of autonomous synthesis robots [19, 20, 190, 191], data on synthesizability and reaction speeds could be collected at the same time with a much higher throughput. Such data will be valuable for decorrelating selection bias and developing broadly applicable synthesis condition predictors.

5.7 Conclusion

In this Chapter, we have developed an interpretable ML method for predicting solid-state synthesis heating temperatures and times on over 6300 reactions synthesis reactions, which are from a larger (over 30,000) synthesis dataset text-mined from scientific literature [68]. The goodness-of-fit values are $R^2 \sim 0.5 - 0.6$ for temperature prediction and

$R^2 \sim 0.3$ for time prediction. For heating temperature prediction, which is an important parameter for solid-state synthesis, the prediction MAE of our model is $\sim 140°\mathrm{C}$. Heating time prediction has a MAE of $\sim 0.3 \log_{10}(h^{-1})$, which translates to a prediction range $[0.5t, 2t]$ if the predicted time is t.

Analysis of the ML models reveals that melting points and formation energies of precursors are good predictors for heating temperatures. By training linear models, we extend Tamman's rule from intermetallics to oxide compounds, which has a well-defined formula for predicting heating temperatures as linearly proportional to the average precursor melting points. Experimentalists may use this extended Tamman's rule to set quick, yet reasonable, initial heating temperatures for new solid-state reactions. The maps of compositional effects (Fig. 5.4) can be further used as guides to choose synthesis conditions with better accuracy given the chemistries of interest. Our model was trained and validated on a diverse set of materials and thus has broad applicability. Moreover, the ML methodologies developed in this work can be applied for learning synthesis conditions on other large synthesis datasets, such as solution-based synthesis of inorganic compounds and nanoparticles [192, 193], or even other tasks where strong model interpretability is preferred.

5.8 Methods

Curation of synthesis training data

We used the dataset of text-mined synthesis recipes that consists of 30,004 solid-state synthesis records [68] to generate the TMR dataset. We took the synthesis conditions of the last heating step in the experimental procedures as the prediction target. The synthesis heating temperatures were predicted in degrees Celsius. The reported heating times were transformed to $\log_{10}(1/t)$ which is not only a better variable for measuring reaction speed, but also shows smaller skewness and long tailedness, which is better predicted by statistical ML models [176]. Note that the TMR dataset is extracted using ML models and contains errors in synthesis conditions. Based on manual inspection, about 5% of the heating temperatures and 16% of the heating times were incorrectly extracted.

To pre-process the dataset, we first removed all entries with no extracted synthesis heating temperatures and times. To obtain thermodynamic data for all targets, we utilized the Materials Project (MP) database [7]. For targets that appear as entries in MP, we simply used the reported thermodynamic information. For targets without a direct match to an MP entry, we performed interpolation by representing them using linear combinations of the most similar entries in the MP as measured by the difference in composition. The 0 K thermodynamic data was then transformed to finite-temperature Gibbs free energies of formation using the previously developed method [194].

Using the finite-temperature $\Delta G_f(T)$ predictions and thermodynamic properties of gases, we computed reaction driving forces, i.e., the grand potential change for the synthesis reactions, $\Delta\Phi_{rxn}$, by assuming the system is open to atmospheric partial pressures of O_2 and CO_2 [197]. The reactions were then decomposed into phase evolution steps by selecting pairs of reactants with the largest grand potential change in each step. Details of the thermodynamic quantity calculation and phase evolution construction can be found and reproduced using the provided codes.

We removed the reactions that cannot be handled by the above thermodynamic calculations (e.g., missing relevant MP entries or containing gases other than O_2 and CO_2), leading to 7,562 remaining reactions. Due to the release of CO_2 gases in carbonate precursor materials, the reaction driving forces have very different distributions for reactions with and without carbonate precursors. Linear models are known to be incapable of properly accounting for the different distributions within groups [179, 180]. Therefore, in our analysis, we split the dataset into carbonate reactions and non-carbonate reactions.

The original PCD collection is semi-structured containing chemical formulas of input/output materials and a natural language description of the synthesis procedure. We used the same approach as in the generation of the TMR dataset to balance synthesis reactions and calculate phase evolution reaction thermodynamic driving forces. The synthesis procedure description text is used to text-mine synthesis operations that contain synthesis condition values. To make the PCD dataset have similar chemistry distribution as the TMR dataset, we only kept oxide syntheses as the TMR dataset is dominated by oxide syntheses. We also ensured there are no duplicates by removing any entries in the PCD dataset that are also in the TMR dataset by matching their article DOIs.

Features for synthesis prediction

For each reaction in the curated training data, we computed four types of synthesis features (133 features in total).

Precursor compound properties. The first type of features (12 in total) are the average/ minimum/ maximum/ difference of melting points, standard enthalpy of formation ΔH_f^{300K}, standard Gibbs free energy of formation ΔG_f^{300K} of precursors. The melting points were retrieved from the NIST Chemistry WebBook [3] and PubChem databases [198], while the thermodynamic properties were retrieved from the FREED database [4], an electronic compilation of the U.S. Bureau of Mines (USBM) thermodynamic data obtained with experiment.

[3] `https://webbook.nist.gov/chemistry/`

[4] `https://www.thermart.net/freed-thermodynamic-database/`

Target compound compositional features. The second type of features are 74 indicator variables representing the presence (1) or absence (0) of different chemical elements in the target compound.

Reaction thermodynamics features. We used 33 thermodynamic features, including the total reaction driving force $\Delta\Phi_{rxn}$, first and last pairwise reaction driving force $\Delta\Phi_{rxn,1}$, $\Delta\Phi_{rxn,-1}$, and the ratio between first/last pairwise reaction driving force and the total reaction driving force, evaluated at different temperatures $T = 800, 900, 1000, 1100, 1200,$ and $1300\,^{\circ}\mathrm{C}$. We also calculated the slope of $\Delta\Phi_{rxn}, \Delta\Phi_{rxn,1}$, and $\Delta\Phi_{rxn,-1}$ by assuming they are linear with respect to temperature and used the slopes as additional features.

Experiment-adjacent features. The fourth type of features are 14 experiment-adjacent features, i.e., indicator variables representing whether certain devices (zirconia balls for ball-milling), experimental procedures (sintering, ball-milling, multiple heating steps, homogenization, repeated grinding, diameter measurement, polycrystalline preparation), and additives (binder materials, distilled water and other liquid additives, phosphors, polyvinyl alcohol) were used in the synthesis.

Since we used WLS which is sensitive to outliers, we performed outlier detection algorithms on the feature values and removed around 10% of reactions. The final training data consists of two datasets totaling 6325 reactions. The subset of carbonate reactions consists of 3,182 reactions. The subset of non-carbonate reactions consists of 3,143 reactions.

Training and evaluation of ML models

We used linear and non-linear regressors to train the ML models. For linear models, we used WLS, a weighted version of ordinary least squares in Python packages *scikit-learn* [157] and *statsmodels* [181]. For non-linear models, we used the XGBoost package [182] and trained GBRT models. To evaluate model goodness-of-fit, we used the coefficient of determination, R-squared (or R^2). For non-linear regressors and out-of-sample evaluations, R^2 is poorly defined and Efron's extended version [199] of Pseudo-R^2 was used. Pseudo-R^2 is calculated as $1-(\mathrm{Mean\ Square\ Error}/\mathrm{Variance\ of\ data})$ and directly comparable to R^2 values.

We implemented DI analysis, a model-agnostic method that calculates the average increase of model R^2 to rank features according to their contribution of predictive powers. Three types of DI values, APDI values, IDI values, and IADI values were computed according to Azen & Budescu [177]. However, to compute the exact APDI values for all the 133 features, we needed to train 2^{133} (sub-)models, which is a computationally prohibitive task. Instead, we estimated APDI values $\overline{\Delta(R^2)}$ by randomly sampling 200 submodels for each feature. All the features were ranked according to the sum of APDI, IDI, and IADI values. This ranking measures the relative predictive powers of the features and was used to sort all features in to an ordered list, as in Fig. 5.2.

We next used the ranking of predictive power to perform forward feature selection for the ML models. Specifically, we started with a linear model with no features but the intercept term. Features were sequentially added into the linear model according to the ranking of predictive power. In this process, we calculated the BIC value of the linear models and removed any feature that would increase the BIC value (an indicator of overfitting). The final list of features were then used in training the models in Fig. 5.5 and Fig. 5.6.

We performed LOOCV to cross-validate regressors and detect overfitting. To test model generalizability, we applied out-of-sample prediction by evaluating model performances on another synthesis conditions dataset compiled from the PCD dataset [178].

Code availability

All codes and data needed to reproduce the results can be found at this repository: `https://github.com/CederGroupHub/s4`.

Chapter 6

Conclusions and outlooks

6.1 Conclusions of this thesis

This thesis demonstrated an ML approach to the prediction of solid-state synthesis by fulfilling two major objectives: 1) constructing a text-mining and NLP pipeline that extracts and codifies solid-state synthesis datasets from the scientific literature, and 2) implementing an interpretable ML method to predict solid-state synthesis conditions (heating temperature and heating time).

In **Chapter 2**, we reviewed key components in a text-mining and IR pipeline that are necessary for working with scientific literature. We introduced practical methods in developing domain-specific NLP and IR tools while emphasizing challenges and opportunities that are unique to materials synthesis literature. In **Chapter 3**, we demonstrated a semi-supervised framework, where unsupervised algorithms leverage the learning of special language patterns in materials synthesis literature and produce features for efficient supervised training. This semi-supervised framework addresses the lack of annotated training data, and was used to develop an accurate classifier that identifies rare synthesis paragraphs. In **Chapter 4**, we introduced our infrastructure for extracting a solid-state synthesis dataset by implementing and combining MER, synthesis operations extraction, and chemical parsing. Our dataset is the first largest machine-readable collection of solid-state synthesis reactions that contains detailed experimental procedure and synthesis conditions. Built upon this dataset, In **Chapter 5**, we developed an ML approach that learns synthesis rules from past experimental results and predicts solid-state synthesis heating temperatures and times. The good interpretability of our ML models allows us to understand how ML predicts synthesis conditions. We discovered that synthesis heating temperatures are highly correlated with precursor material stability and heating times are highly correlated with experimental setups/intentions. We also extended Tamman's rule on intermetallics synthesis temperature to ionic systems and quantitatively evaluated the prediction performances of this empirical rule.

The biggest challenge in these works, is to keep a balance between collecting a wide coverage of synthesis data and curating a smaller subset in which data analysis and modeling yield tangible synthesis rules or insights. For example, we scraped a scientific text corpus containing 5.4 million papers from more than 10 publishers. It is very easy to apply massive data analysis methods to all the papers, but many synthesis rules only work for small sets of chemical systems, which can be easily overlooked in these massive analyses. Instead, in the works described in this thesis, we start with a large paper collection but gradually shift focus on one particular synthesis method - solid-state synthesis.

Focusing on the topic of solid-state synthesis, this thesis described a few smaller projects that are built on top of each other to achieve the goal of synthesis prediction:

- Starting with 5.4 million papers, we developed a synthesis classifier (Chapter 3) that identified ~ 100K solid-state synthesis paragraphs. All non-relevant paragraphs, such as papers in other fields or paragraphs that do not describe synthesis procedures, were removed from this stage.
- Using the ~ 100K solid-state synthesis paragraphs, we performed NER to identify relevant elements of a synthesis, and put them together into a dataset of 30K complete synthesis reactions. We also recognize and parse the list of experimental procedures and conditions (Chapter 4).
- This dataset of 30K synthesis reactions was then used as training data to train ML models (Chapter 5) that predict two important synthesis conditions: temperature and time. Text-mined data become less relevant in this stage, and more explicit modeling of the chemical reaction thermodynamics are required. Therefore, we heavily relied on our domain knowledge of solid-state synthesis to create features in ML models.

Put into a broader context, the works in this thesis only constitute a small proportion of text-mining and ML in the field of materials synthesis literature. There are so many unsolved problems and exciting new areas of research, both inspired by the developments of the general NLP field and the materials synthesis science community. Finally, in this Chapter, we outline some interesting topics that could be addressed in future research.

6.2 Future works

Opportunities enabled by large pretrained language models

In more recent years, the NLP community has shifted heavily towards very deep neural network language models *pretrained* on unlabeled large corpora, particularly Transformers [104] and BERT [105]. These models use self-supervised training methods such as

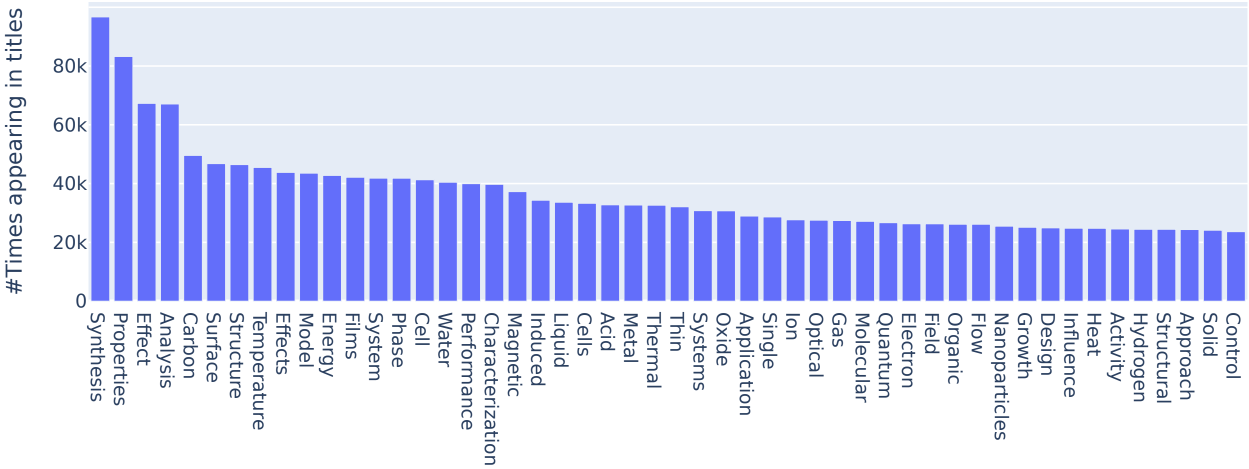


Figure 6.1: Frequencies of the keywords appearing in the materials synthesis corpus, which was used to train MatBERT.

masked language modeling [104] and causal language modeling [200]. Training on massive unlabeled text corpus has enabled these models to learn language patterns with more complex features, enabling few-shot learning [107, 201] or even zero-shot learning [201], where new prediction tasks could be trained with very little labels or even no labels.

The NLP opportunities enabled by such pretrained language models could be certainly leveraged in materials informatics. Using full-text of the 5.4 million papers, we were able to pretrain MatBERT [1], a BERT model on materials science literature. Fig. 6.1 demonstrates the frequencies of the vocabulary appearing in the titles of the pre-training corpus. MatBERT specializes in understanding materials science terminologies and paragraph-level scientific reasoning. With the specialized pretraining corpus, our initial study has revealed MatBERT can perform zero-shot learning of synthesis classification, as demonstrated below.

> Input paragraph: α-Fe2O3 used in present study was prepared by a combination of precipitation and spray-drying technologies. In brief, a solution containing Fe(NO3)3•9H2O was used in precipitation with NH4OH solution as a precipitator at pH=8.5 9.0 and T=70 °C. The precipitate was washed and then filtered. The mixture was reslurried and spray-dried. Finally, a sample with diameters of 20 26μm was calcined at 450°C for 5 h in a muffle furnace.
>
> Append prompt and predict: Materials used in this study were prepared by the conventional [MASK] method.

[1] https://github.com/lbnlp/MatBERT

Answer by MatBERT:
P(MASK = coprecipitation) = 0.52,
P(MASK = precipitation) = 0.25,
P(MASK = hydrothermal) = 0.03

In the above example, a plain text paragraph is appended with a *prompt*, which contains a short sentence and a special [MASK] token to be filled in by MatBERT. The paragraph and the prompt are sent to MatBERT to perform the masked language modeling prediction task. MatBERT produces a distribution over different words for [MASK], where "coprecipitation" and "precipitation" have the highest probability, matching the synthesis method used in this paragraph. MatBERT learns how to fill in the [MASK] blank token by training itself on a large number of existing paragraphs.

This zero-shot learning of synthesis classification is only one of the many things we could potentially achieve with MatBERT. For example, MatBERT is being used to train better NER models [87]. In the future, other promising applications, such as **relation extraction** [91, 112, 120], **sentence entailment** (determining the logical/semantic relationship between sentences) [202], **open-domain question/answering** [203–205], could also be applied to materials synthesis corpus for better modeling and utilization of synthesis knowledge and insights in the literature.

Recognizing synthesis reactions using reinforcement learning

In Chapter 2 and Chapter 4, we reviewed possible methods and demonstrated our implementation of the MER problem. One of the primary goals of MER is to identify solid-state synthesis reactions, such as $\frac{1+x}{2}\mathrm{Li_2CO_3}+(2-x)\mathrm{MnO2}+\frac{3x-1}{4}\mathrm{O_2}=\mathrm{Li_{1+x}Mn_{2-x}O_4}+\frac{1+x}{2}\mathrm{CO_2}$, which is extracted from the sample paragraph in Fig. 6.2. In Chapter 4 and He *et al.* [83], we implemented MER using an extended version of the NER model trained on annotated word tags. Many other works on materials science NER [83, 87, 101] have also followed the same practice. However, we note there are a few disadvantages of directly applying NER models to synthesis reaction extraction:

1. NER models require a certain amount of annotation data for training, while the amount of annotated data remains very limited in materials informatics. Moreover, NER has degraded performance when applied to different distributions of text [64, 183].

2. In NER models, the recognition of each material is independent (or partially dependent) of other materials. Thus, the success rate of finding fully balanced reactions with N materials decreases exponentially as p^N if the accuracy of each material is p. To achieve accurate extraction of balanced reactions, one needs to have extremely accurate NER models, which in turn requires more annotated training data.

> Li1+xMn2-xO4 spinels were fabricated by solid-state synthesis. Stoichiometric amounts of Li2CO3 and MnO2 were mixed thoroughly in an agate. Li1+xMn2-xO4-zFz samples were made in a similar manner utilizing LiF as a precursor.

Figure 6.2: Sample paragraph for MER which contains two distinct solid-state synthesis reactions. Precursor materials are marked in blue, and target materials are marked in red. A RL method may be able to reinforce the learning of the tags of precursor/target materials by leveraging the conservation of chemical elements. For example, material MnO2 may be tagged as precursor material by already knowing Li1+xMn2-xO4 is the target material and Li2CO3 is one of the precursor materials.

The recognition of reaction entities, i.e., MER tasks, is fundamentally different from NER. In NER, there are no obvious relations among the entities. In the example of "Barack Obama is the 44th president of the U.S.", we do not need to inspect other named entities to declare that "Barack Obama" is a person. However, there exists a strong relational constraint between target and precursor materials in MER. For example, in the sample paragraph in Fig. 6.2, if we have identified target material $\mathrm{Li_{1+x}Mn_{2-x}O_4}$ and precursor material $\mathrm{Li_2CO_3}$, then by conservation of chemical elements, another precursor material must contain element Mn. This information can be used to help identify another precursor material $\mathrm{MnO_2}$.

One approach to leverage this constraint from the conservation of chemical elements is to apply reinforcement learning (RL) [206]. In RL, the problem of MER is transformed into a decision-making *environment* where an *agent* needs to tag the type of each word according to the paragraph context and the current labels of the words, in order to maximize the *reward*. The reward function is designed in the way that the agent experiences the maximal reward if a balanced reaction is identified and gets penalized if a word is wrongly tagged as precursor or target materials. The identification of balanced reaction can be easily realized, for example by using the *MaterialsParser* and *ReactionCompleter* codes (in Chapter 4) to parse the chemical formulas and attempt balance reaction equations.

Training MER models using RL has two main advantages. First, the agent is able to explore possible solutions by performing random actions by itself until a balanced equation is identified. In other words, the agent is able to *learn by trying*, without relying on any annotated data. Second, the agent can be made *online* [206]. In an online algorithm, if the agent could not produce the right set of tags in initial attempts, it is allowed to continue the exploration until the balanced reaction is found. In this way, the MER model can *adapt to new texts* by interacting with the paragraphs and the reward function.

Improving synthesis predictors for practical solid-state synthesis design

In Chapter 5, we framed the prediction of solid-state synthesis conditions (heating temperature and time) as regression problems and achieved reasonable prediction performances ($R^2 \sim 0.5 - 0.6$ for heating temperature prediction and $R^2 \sim 0.3$ for heating time prediction). However, these performances are still far from being practical such that they can be deployed in synthesis labs. Here, we summarize a few important aspects for increasing model performance as potential improvements in the future.

Better synthesis features. Features are limiting factors in creating ML models with high predictive power. The work in Chapter 5 used 133 features spanning four categories: precursor material properties, target material compositions, reaction thermodynamics, and experiment-adjacent features. Besides these features, one set of useful features may be further factors that indicate the intention of syntheses. For example, desired microstructure of the target materials (single-crystal or spin-coated materials). These features are expressed in papers in more subtle ways and hard to directly extract. One workaround to learn from these features is to find correlated variables that are easier to extract, such as applications of the target compound (battery materials v.s. thermoelectric materials). With the advancing NLP techniques [87, 101] for materials science and chemistry, the features such as desired microstructures may be directly extracted with higher qualities and reveal important aspects of synthesis experiments.

Improved NLP data collection. Due to the probabilistic nature of the text-mining pipeline that extracted the datasets (Chapter 4), errors in the training data are inevitable [68]. Manual inspection of the dataset reveals that 95% of the extracted synthesis heating temperature values and 84% of the extracted heating time values are correct. Improved text-mining algorithms can thus improve data quality and increase ML model performance.

Modeling non-uniqueness. In Chapter 5, we modeled synthesis condition predictions as point value regression problems. However, this may be sub-optimal, as the conditions where a given synthesis can proceed are non-unique and often span a range of values. Consequently, there is not a unique ground truth of optimal synthesis conditions, which brings irreducible error to ML models. The issue of non-uniqueness is even more problematic for heating time prediction. If the synthesis finishes within t_0, then any heating time $t > t_0$ will yield the desired compound if it is thermodynamically stable at the synthesis conditions. As a result, heating time is seldom optimized but based heavily on furnace heating schedule, lab shifts, etc. Indeed, in Fig. 5.5, our ML models have a larger error for predicting heating time than heating temperature. Modeling synthesis conditions as distributions, e.g., generalized linear models [207], could in principle solve this issue. Such techniques not only require more data to train models, but also need different evaluation methods to properly validate the models.

Negative samples. Negative experimental results are rarely reported in papers. Nevertheless, from an ML point of view, negative data are extremely useful for learning the exact decision boundaries of synthesis conditions. Besides, negative data can be used in other classification tasks, such as predicting the type of synthesis techniques, heating atmospheres, etc. Unfortunately, negative experimental results are rarely reported in the literature or described in very subtle languages. In future works, negative samples may be collected using two approaches. First, with the promising state-of-the-art NLP methods (described in Section 6.2), we may be able to text-mine "sentiment" or "entailment" scores for synthesis outcomes and associate them with the extracted synthesis data entries. Second, recent developments of autonomous synthesis robots [19, 20, 190, 191] may be directly used to collect synthesis data with much higher throughput and better quality (in terms of controlled experimental conditions), which could be ultimately used to train prediction models with better accuracy.

Bibliography

1. Curtarolo, S., Hart, G. L. W., Nardelli, M. B., Mingo, N., Sanvito, S. & Levy, O. The high-throughput highway to computational materials design. *Nature Materials* **12,** 191–201 (2013).
2. Jain, A., Shin, Y. & Persson, K. A. Computational predictions of energy materials using density functional theory. *Nature Reviews Materials* **1,** 1–13 (2016).
3. Schleder, G. R., Padilha, A. C. M., Acosta, C. M., Costa, M. & Fazzio, A. From DFT to machine learning: recent approaches to materials science–a review. *Journal of Physics: Materials* **2,** 032001 (May 2019).
4. Butler, K. T., Davies, D. W., Cartwright, H., Isayev, O. & Walsh, A. Machine learning for molecular and materials science. *Nature* **559,** 547–555 (2018).
5. Lookman, T., Balachandran, P. V., Xue, D. & Yuan, R. Active learning in materials science with emphasis on adaptive sampling using uncertainties for targeted design. *npj Computational Materials* **5,** 21 (2019).
6. Rickman, J. M., Lookman, T. & Kalinin, S. V. Materials informatics: From the atomic-level to the continuum. *Acta Materialia* **168,** 473–510 (2019).
7. Jain, A., Ong, S. P., Hautier, G., Chen, W., Richards, W. D., Dacek, S., Cholia, S., Gunter, D., Skinner, D., Ceder, G., *et al.* Commentary: The Materials Project: A materials genome approach to accelerating materials innovation. *APL Mater* **1,** 011002 (2013).
8. Saal, J. E., Kirklin, S., Aykol, M., Meredig, B. & Wolverton, C. Materials design and discovery with high-throughput density functional theory: The Open Quantum Materials Database (OQMD). *JOM* **65,** 1501–1509. ISSN: 1543-1851 (Nov. 2013).
9. Curtarolo, S., Setyawan, W., Wang, S., Xue, J., Yang, K., Taylor, R. H., Nelson, L. J., Hart, G. L., Sanvito, S., Buongiorno-Nardelli, M., Mingo, N. & Levy, O. AFLOWLIB.ORG: A distributed materials properties repository from high-throughput ab initio calculations. *Computational Materials Science* **58,** 227–235. ISSN: 0927-0256 (2012).
10. Draxl, C. & Scheffler, M. NOMAD: The FAIR concept for big data-driven materials science. *MRS Bulletin* **43,** 676–682 (2018).

11. O'Mara, J., Meredig, B. & Michel, K. Materials data infrastructure: A case study of the Citrination platform to examine data import, storage, and access. *JOM* **68,** 2031–2034. ISSN: 1543-1851 (2016).

12. Aykol, M., Dwaraknath, S. S., Sun, W. & Persson, K. A. Thermodynamic limit for synthesis of metastable inorganic materials. *Science advances* **4,** eaaq0148 (2018).

13. Sun, W., Dacek, S. T., Ong, S. P., Hautier, G., Jain, A., Richards, W. D., Gamst, A. C., Persson, K. A. & Ceder, G. The thermodynamic scale of inorganic crystalline metastability. *Science advances* **2,** e1600225 (2016).

14. Jain, A., Hautier, G., Moore, C., Kang, B., Lee, J., Chen, H., Twu, N. & Ceder, G. A computational investigation of Li9M3 (P2O7) 3 (PO4) 2 (M= V, Mo) as cathodes for Li ion batteries. *Journal of The Electrochemical Society* **159,** A622 (2012).

15. Wu, Y. & Ceder, G. First principles study on Ta3N5: Ti3O3N2 solid solution as a water-splitting photocatalyst. *The Journal of Physical Chemistry C* **117,** 24710–24715 (2013).

16. Sun, W. Q., Wolverton, C., Akbarzadeh, A. & Ozolins, V. First-principles prediction of high-capacity, thermodynamically reversible hydrogen storage reactions based on (NH 4) 2 B 12 H 12. *Physical Review B* **83,** 064112 (2011).

17. Chen, H., Hautier, G., Jain, A., Moore, C., Kang, B., Doe, R., Wu, L., Zhu, Y., Tang, Y. & Ceder, G. Carbonophosphates: a new family of cathode materials for Li-ion batteries identified computationally. *Chemistry of Materials* **24,** 2009–2016 (2012).

18. Zhu, H., Hautier, G., Aydemir, U., Gibbs, Z. M., Li, G., Bajaj, S., Pöhls, J.-H., Broberg, D., Chen, W., Jain, A., *et al.* Computational and experimental investigation of TmAgTe2 and XYZ2 compounds, a new group of thermoelectric materials identified by first-principles high-throughput screening. *Journal of Materials Chemistry C* **3,** 10554–10565 (2015).

19. Szymanski, N., Zeng, Y., Huo, H., Bartel, C., Kim, H. & Ceder, G. Toward autonomous design and synthesis of novel inorganic materials. *Materials Horizons* (2021).

20. Kimmig, J., Zechel, S. & Schubert, U. S. Digital transformation in materials science: A paradigm change in material's development. *Advanced Materials* **33,** 2004940 (2021).

21. Rao, C. N. R. & Biswas, K. *Essentials of inorganic materials synthesis* (John Wiley & Sons, 2015).

22. Kohlmann, H. Looking into the black box of solid-State synthesis. *European Journal of Inorganic Chemistry* **2019,** 4174–4180 (2019).

23. Chamorro, J. R. & McQueen, T. M. Progress toward solid state synthesis by design. *Accounts of chemical research* **51,** 2918–2925 (2018).

24. Shoemaker, D. P., Hu, Y.-J., Chung, D. Y., Halder, G. J., Chupas, P. J., Soderholm, L., Mitchell, J. & Kanatzidis, M. G. In situ studies of a platform for metastable inorganic crystal growth and materials discovery. *Proceedings of the National Academy of Sciences* **111,** 10922–10927 (2014).

25. McClain, R., Malliakas, C. D., Shen, J., He, J., Wolverton, C., González, G. B. & Kanatzidis, M. G. Mechanistic insight of KBiQ 2 (Q= S, Se) using panoramic synthesis towards synthesis-by-design. *Chemical Science* **12,** 1378–1391 (2021).

26. Ito, H., Shitara, K., Wang, Y., Fujii, K., Yashima, M., Goto, Y., Moriyoshi, C., Rosero-Navarro, N. C., Miura, A. & Tadanaga, K. Kinetically stabilized cation arrangement in Li3YCl6 superionic conductor during solid-State reaction. *Advanced Science,* 2101413 (2021).

27. Paradis-Fortin, L., Lemoine, P., Prestipino, C., Kumar, V. P., Raveau, B., Nassif, V., Cordier, S. & Guilmeau, E. Time-resolved in situ neutron diffraction study of Cu22Fe8 Ge4S32 germanite: a guide for the synthesis of complex chalcogenides. *Chemistry of Materials* **32,** 8993–9000 (2020).

28. Bianchini, M., Wang, J., Clément, R. J., Ouyang, B., Xiao, P., Kitchaev, D., Shi, T., Zhang, Y., Wang, Y., Kim, H., *et al.* The interplay between thermodynamics and kinetics in the solid-state synthesis of layered oxides. *Nature materials* **19,** 1088–1095 (2020).

29. Chen, B.-R., Sun, W., Kitchaev, D. A., Mangum, J. S., Thampy, V., Garten, L. M., Ginley, D. S., Gorman, B. P., Stone, K. H., Ceder, G., *et al.* Understanding crystallization pathways leading to manganese oxide polymorph formation. *Nature communications* **9,** 1–9 (2018).

30. Jiang, Z., Ramanathan, A. & Shoemaker, D. P. In situ identification of kinetic factors that expedite inorganic crystal formation and discovery. *Journal of Materials Chemistry C* **5,** 5709–5717 (2017).

31. Martinolich, A. J. & Neilson, J. R. Toward reaction-by-design: achieving kinetic control of solid state chemistry with metathesis. *Chemistry of Materials* **29,** 479–489 (2017).

32. Miura, A., Bartel, C. J., Goto, Y., Mizuguchi, Y., Moriyoshi, C., Kuroiwa, Y., Wang, Y., Yaguchi, T., Shirai, M., Nagao, M., *et al.* Observing and modeling the sequential pairwise reactions that drive solid-State ceramic synthesis. *Advanced Materials,* 2100312 (2021).

33. Miura, A., Ito, H., Bartel, C. J., Sun, W., Rosero-Navarro, N. C., Tadanaga, K., Nakata, H., Maeda, K. & Ceder, G. Selective metathesis synthesis of MgCr 2 S 4 by control of thermodynamic driving forces. *Materials horizons* **7,** 1310–1316 (2020).

34. McDermott, M. J., Dwaraknath, S. S. & Persson, K. A. A graph-based network for predicting chemical reaction pathways in solid-state materials synthesis. *Nature communications* **12,** 1–12 (2021).

35. Aykol, M., Montoya, J. H. & Hummelshøj, J. Rational solid-state synthesis routes for inorganic materials. *Journal of the American Chemical Society* (2021).

36. Sun, W., Holder, A., Orvañanos, B., Arca, E., Zakutayev, A., Lany, S. & Ceder, G. Thermodynamic routes to novel metastable nitrogen-Rich nitrides. *Chemistry of Materials* **29,** 6936–6946 (2017).

37. Wustrow, A., Huang, G., McDermott, M. J., O'Nolan, D., Liu, C.-H., Tran, G. T., McBride, B. C., Dwaraknath, S. S., Chapman, K. W., Billinge, S. J., *et al.* Lowering ternary oxide synthesis temperatures by solid-State cometathesis reactions. *Chemistry of Materials* (2021).

38. Sun, W., Jayaraman, S., Chen, W., Persson, K. A. & Ceder, G. Nucleation of metastable aragonite CaCO3 in seawater. *Proceedings of the National Academy of Sciences* **112,** 3199–3204 (2015).

39. Segler, M. H. S., Preuss, M. & Waller, M. P. Planning chemical syntheses with deep neural networks and symbolic AI. *Nature* **555,** 604–610 (2018).

40. Feng, F., Lai, L. & Pei, J. Computational chemical synthesis analysis and pathway design. *Frontiers in chemistry* **6,** 199 (2018).

41. Wei, J. N., Duvenaud, D. & Aspuru-Guzik, A. Neural networks for the prediction of organic chemistry reactions. *ACS central science* **2,** 725–732 (2016).

42. Duvenaud, D. K., Maclaurin, D., Iparraguirre, J., Bombarell, R., Hirzel, T., Aspuru-Guzik, A. & Adams, R. P. in *Advances in Neural Information Processing Systems 28* (eds Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M. & Garnett, R.) 2224–2232 (Currant Associates, Inc., 2015).

43. Ley, S. V., Fitzpatrick, D. E., Ingham, R. J. & Myers, R. M. Organic synthesis: March of the machines. *Angewandte Chemie International Edition* **54,** 3449–3464 (2015).

44. Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žıdek, A., Potapenko, A., *et al.* Highly accurate protein structure prediction with AlphaFold. *Nature* **596,** 583–589 (2021).

45. Senior, A. W., Evans, R., Jumper, J., Kirkpatrick, J., Sifre, L., Green, T., Qin, C., Žıdek, A., Nelson, A. W., Bridgland, A., *et al.* Improved protein structure prediction using potentials from deep learning. *Nature* **577,** 706–710 (2020).

46. Goodman, J. Computer Software Review: Reaxys. *Journal of Chemical Information and Modeling* **49,** 2897–2898 (12 2009).

47. Kim, S., Chen, J., Cheng, T., Gindulyte, A., He, J., He, S., Li, Q., Shoemaker, B. A., Thiessen, P. A., Yu, B., Zaslavsky, L., Zhang, J. & Bolton, E. E. PubChem 2019 update: improved access to chemical data. *Nucleic Acids Research* **47,** D1102–D1109 (2018).

48. Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N. & Bourne, P. E. The protein data bank. *Nucleic acids research* **28,** 235–242 (2000).

49. Mirdita, M., von den Driesch, L., Galiez, C., Martin, M. J., Söding, J. & Steinegger, M. Uniclust databases of clustered and deeply annotated protein sequences and alignments. *Nucleic acids research* **45,** D170–D176 (2017).

50. Raccuglia, P., Elbert, K. C., Adler, P. D. F., Falk, C., Wenny, M. B., Mollo, A., Zeller, M., Friedler, S. A., Schrier, J. & Norquist, A. J. Machine-learning-assisted materials discovery using failed experiments. *Nature* **533,** 73–76 (2016).

51. Xu, R. J., Olshansky, J. H., Adler, P. D., Huang, Y., Smith, M. D., Zeller, M., Schrier, J. & Norquist, A. J. Understanding structural adaptability: a reactant informatics approach to experiment design. *Molecular Systems Design & Engineering* **3,** 473–484 (2018).

52. Belsky, A., Hellenbrandt, M., Karen, V. L. & Luksch, P. New developments in the Inorganic Crystal Structure Database (ICSD): accessibility in support of materials research and design. *Acta Crystallographica Section B: Structural Science* **58,** 364–369 (2002).

53. *NIST Chemistry WebBook, NIST Standard Reference Database Number 69* (eds Linstrom, P. & Mallard, W.) (National Institute of Standards and Technology, Gaithersburg MD, 20899, 2019).

54. Blokhin, E. & Villars, P. in, 1–26 (Springer, Cham, 2018).

55. Villars, P. & Cenzual, K. *Pearson's Crystal Data: Crystal Structure Database for Inorganic Compounds (on DVD)* Materials Park, Ohio, USA, 2018.

56. Bornmann, L. & Mutz, R. Growth rates of modern science: A bibliometric analysis based on the number of publications and cited references. *Journal of the Association for Information Science and Technology* **66,** 2215–2222 (2015).

57. Ware, M. & Mabe, M. An overview of scientific and scholarly journal publishing. *The STM report* (2009).

58. Krallinger, M., Rabal, O., Lourenço, A., Oyarzabal, J. & Valencia, A. Information retrieval and text mining technologies for chemistry. *Chemical Reviews* **117,** 7673–7761 (2017).

59. Swain, M. C. & Cole, J. M. ChemDataExtractor: a toolkit for automated extraction of chemical information from the scientific literature. *Journal of chemical information and modeling* **56,** 1894–1904 (2016).

60. Jessop, D. M., Adams, S. E., Willighagen, E. L., Hawizy, L. & Murray-Rust, P. OSCAR4: a flexible architecture for chemical text-mining. *Journal of Cheminformatics* **3,** 41 (2011).

61. Hawizy, L., Jessop, D. M., Adams, N. & Murray-Rust, P. ChemicalTagger: A tool for semantic text-mining in chemistry. *Journal of Cheminformatics* **3,** 1–13 (2011).

62. Eltyeb, S. & Salim, N. Chemical named entities recognition: a review on approaches and applications. *Journal of cheminformatics* **6,** 1–12 (2014).

63. Kim, E., Huang, K., Tomala, A., Matthews, S., Strubell, E., Saunders, A., McCallum, A. & Olivetti, E. Machine-learned and codified synthesis parameters of oxide materials. *Scientific Data* **4,** 170127 (2017).

64. Kononova, O., He, T., Huo, H., Trewartha, A., Olivetti, E. A. & Ceder, G. Opportunities and challenges of text mining in materials research. *iScience* **24** (2021).

65. Miner, G., Elder IV, J., Fast, A., Hill, T., Nisbet, R. & Delen, D. *Practical text mining and statistical analysis for non-structured text data applications* ISBN: 9780123870117 (Elsevier Science, 2012).

66. Jurafsky, D. *Speech & language processing* (Pearson Education India, 2000).

67. Kolárik, C., Klinger, R., Friedrich, C. M., Hofmann-Apitius, M. & Fluck, J. *Chemical Names: Terminological resources and corpora annotation* in *Workshop on Building and evaluating resources for biomedical text mining* (2008), 51–58.

68. Kononova, O., Huo, H., He, T., Rong, Z., Botari, T., Sun, W., Tshitoyan, V. & Ceder, G. Text-mined dataset of inorganic materials synthesis recipes. *Scientific data* **6,** 1–11 (2019).

69. Trewartha, A., Dagdelen, J., Huo, H., Cruse, K., Wang, Z., He, T., Subramanian, A., Fei, Y., Justus, B., Persson, K. & Ceder, G. *COVIDScholar: An automated COVID-19 research aggregation and analysis platform* 2020. arXiv: `2012.03891 [cs.DL]`.

70. Constantin, A., Pettifer, S. & Voronkov, A. *PDFX: Fully-Automated PDF-to-XML Conversion of Scientific Literature* in *Proceedings of the 2013 ACM Symposium on Document Engineering* (Association for Computing Machinery, 2013), 177–180.

71. Tkaczyk, D., Szostek, P., Fedoryszak, M., Dendek, P. & Bolikowski, Ł. CERMINE: automatic extraction of structured metadata from scientific literature. *International Journal on Document Analysis and Recognition (IJDAR)* **18,** 317–335 (2015).

72. Luong, M.-T., Nguyen, T. D. & Kan, M.-Y. Logical structure recovery in scholarly articles with rich document features, 270–292 (2012).

73. Mouchere, H., Zanibbi, R., Garain, U. & Viard-Gaudin, C. Advancing the state of the art for handwritten math recognition: the CROHME competitions, 2011–2014. *IJDAR* **19,** 173–189 (2016).

74. Mahdavi, M., Zanibbi, R., Mouchere, H., Viard-Gaudin, C. & Garain, U. *ICDAR 2019 CROHME+ TFD: Competition on recognition of handwritten mathematical expressions and typeset formula detection* in *2019 International Conference on Document Analysis and Recognition (ICDAR)* (2019), 1533–1538.

75. Memon, J., Sami, M., Khan, R. A. & Uddin, M. Handwritten optical character recognition (OCR): a comprehensive systematic literature review (SLR). *IEEE Access* **8,** 142642–142668 (2020).

76. Ramakrishnan, C., Patnia, A., Hovy, E. & Burns, G. A. Layout-aware text extraction from full-text PDF of scientific articles. *Source Code for Biology and Medicine* **7,** 7 (2012).

77. Read, J., Dridan, R., Oepen, S. & Solberg, L. J. *Sentence boundary detection: A long solved problem?* in *Proceedings of COLING 2012: Posters* (2012), 985–994.

78. Leaman, R., Wei, C.-H. & Lu, Z. tmChem: a high performance approach for chemical named entity recognition and normalization. *Journal of Cheminformatics* **7,** S3 (2015).

79. Bird, S., Klein, E. & Loper, E. *Natural language processing with Python: analyzing text with the natural language toolkit* (O'Reilly Media, Inc., 2009).

80. Honnibal, M. & Johnson, M. *An improved non-monotonic transition system for dependency parsing* in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (Association for Computational Linguistics, Lisbon, Portugal, Sept. 2015), 1373–1378.

81. Huang, L. & Ling, C. Representing multiword chemical terms through phrase-level preprocessing and word embedding. *ACS omega* **4,** 18510–18519 (2019).

82. Alperin, B. L., Kuzmin, A. O., Ilina, L. Y., Gusev, V. D., Salomatina, N. V. & Parmon, V. N. Terminology spectrum analysis of natural-language chemical documents: term-like phrases retrieval routine. *Journal of cheminformatics* **8,** 1–17 (2016).

83. He, T., Sun, W., Huo, H., Kononova, O., Rong, Z., Tshitoyan, V., Botari, T. & Ceder, G. Similarity of precursors in solid-state synthesis as text-mined from scientific literature. *Chemistry of Materials* **32,** 7861–7873 (2020).

84. Sennrich, R., Haddow, B. & Birch, A. Neural machine translation of rare words with subword units. arXiv: `1508.07909 [cs.CL]` (2015).

85. Schuster, M. & Nakajima, K. *Japanese and korean voice search* in *2012 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (2012), 5149–5152.

86. Kudo, T. & Richardson, J. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226* (2018).

87. Trewartha, A., Walker, N., Huo, H., Lee, S., Cruse, K., Dagdelen, J., Dunn, A., Persson, K. A., Ceder, G. & Jain, A. Quantifying the advantage of domain-specific pretraining on named entity recognition tasks in materials science. *Patterns* **3,** 100488 (2022).

88. Corbett, P. & Copestake, A. Cascaded classifiers for confidence-based chemical named entity recognition. *BMC Bioinformatics* **9,** S4 (2008).

89. Harris, Z. S. Distributional structure. *Word* **10,** 146–162 (1954).

90. Liu, Z., Lin, Y. & Sun, M. *Representation learning for natural language processing* 1st ed. (Springer Singapore, 2020).

91. Court, C. J. & Cole, J. M. Auto-generated materials database of Curie and Néel temperatures via semi-supervised relationship extraction. *Scientific Data* **5,** 180111 (2018).

92. Hiszpanski, A. M., Gallagher, B., Chellappan, K., Li, P., Liu, S., Kim, H., Han, J., Kailkhura, B., Buttler, D. J. & Han, T. Y.-J. Nanomaterials synthesis insights from machine learning of scientific articles by extracting, structuring, and visualizing knowledge. *Journal of Chemical Information and Modeling* (2020).

93. Blei, D. M. Probabilistic topic models. *Communications of the ACM* **55,** 77–84 (2012).

94. Blei, D. M., Ng, A. Y. & Jordan, M. I. Latent dirichlet allocation. *the Journal of machine Learning research* **3,** 993–1022 (2003).

95. Huo, H., Rong, Z., Kononova, O., Sun, W., Botari, T., He, T., Tshitoyan, V. & Ceder, G. Semi-supervised machine-learning classification of materials synthesis procedures. *npj Computational Materials* **5,** 62 (2019).

96. Pennington, J., Socher, R. & Manning, C. D. *Glove: Global vectors for word representation* in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (Association for Computational Linguistics, Doha, Qatar, 2014), 1532–1543.

97. Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K. & Zettlemoyer, L. *Deep contextualized word representations* in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (Association for Computational Linguistics, New Orleans, Louisiana, 2018), 2227–2237.

98. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. & Dean, J. in *Advances in Neural Information Processing Systems 26* (eds Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z. & Weinberger, K. Q.) 3111–3119 (Curran Associates, Inc., 2013).

99. Bojanowski, P., Grave, E., Joulin, A. & Mikolov, T. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics* **5,** 135–146 (2017).

100. Kim, E., Jensen, Z., van Grootel, A., Huang, K., Staib, M., Mysore, S., Chang, H.-S., Strubell, E., McCallum, A., Jegelka, S., *et al.* Inorganic materials synthesis planning with literature-trained neural networks. *Journal of chemical information and modeling* **60,** 1194–1201 (2020).

101. Weston, L., Tshitoyan, V., Dagdelen, J., Kononova, O., Trewartha, A., Persson, K. A., Ceder, G. & Jain, A. Named entity recognition and normalization applied to large-scale information extraction from the materials science literature. *Journal of chemical information and modeling* **59,** 3692–3702 (2019).

102. Tshitoyan, V., Dagdelen, J., Weston, L., Dunn, A., Rong, Z., Kononova, O., Persson, K. A., Ceder, G. & Jain, A. Unsupervised word embeddings capture latent knowledge from materials science literature. *Nature* **571,** 95–98 (2019).

103. Bahdanau, D., Cho, K. & Bengio, Y. Neural machine translation by jointly learning to align and translate. arXiv: `1409.0473 [cs.CL]` (2016).

104. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. & Polosukhin, I. *Attention is all you need* in *Advances in neural information processing systems* (2017), 5998–6008.

105. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv: `1810.04805 [cs.CL]` (2018).

106. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., *et al.* Language models are unsupervised multitask learners. *OpenAI Blog* **1,** 9 (2019).

107. Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., *et al. Language models are few-shot learners* in (2020). arXiv: `2005.14165 [cs.CL]`.

108. Kuniyoshi, F., Makino, K., Ozawa, J. & Miwa, M. Annotating and extracting synthesis process of all-solid-state batteries from scientific literature. arXiv: `2002.07339 [cs.CL]` (2020).

109. Vaucher, A. C., Zipoli, F., Geluykens, J., Nair, V. H., Schwaller, P. & Laino, T. Automated extraction of chemical synthesis actions from experimental procedures. *Nature Communications* **11,** 3601 (2020).

110. Rocktäschel, T., Weidlich, M. & Leser, U. ChemSpot: a hybrid system for chemical named entity recognition. *Bioinformatics* **28,** 1633–1640 (2012).

111. Garcıa-Remesal, M., Garcıa-Ruiz, A., Pérez-Rey, D., De La Iglesia, D. & Maojo, V. Using nanoinformatics methods for automatically identifying relevant nanotoxicology entities from the literature. *BioMed research international* **2013** (2013).

112. Shah, S., Vora, D., Gautham, B. & Reddy, S. A relation aware search engine for materials science. *Integrating Materials and Manufacturing Innovation* **7,** 1–11 (2018).

113. Tchoua, R. B., Ajith, A., Hong, Z., Ward, L. T., Chard, K., Belikov, A., Audus, D. J., Patel, S., de Pablo, J. J. & Foster, I. T. *Creating training data for scientific named entity recognition with minimal human effort* in *International Conference on Computational Science* (2019), 398–411.

114. Lafferty, J., McCallum, A. & Pereira, F. C. *Conditional Random Fields: Probabilistic models for segmenting and labeling sequence data* in *ICML* (2001).

115. Hochreiter, S. & Schmidhuber, J. Long Short-Term Memory. *Neural Computation* **9,** 1735–1780 (1997).

116. Gurulingappa, H., Mudi, A., Toldo, L., Hofmann-Apitius, M. & Bhate, J. Challenges in mining the literature for chemical information. *Rsc Advances* **3,** 16194–16211 (2013).

117. Olivetti, E. A., Cole, J. M., Kim, E., Kononova, O., Ceder, G., Han, T. Y.-J. & Hiszpanski, A. M. Data-driven materials research enabled by natural language processing and information extraction. *Applied Physics Reviews* **7,** 041317 (2020).

118. Lowe, D. M. & Sayle, R. A. LeadMine: a grammar and dictionary driven approach to entity recognition. *Journal of cheminformatics* **7,** 1–9 (2015).

119. Korvigo, I., Holmatov, M., Zaikovskii, A. & Skoblov, M. Putting hands to rest: efficient deep CNN-RNN architecture for chemical named entity recognition with no hand-crafted rules. *Journal of Cheminformatics* **10,** 28 (2018).

120. Onishi, T., Kadohira, T. & Watanabe, I. Relation extraction with weakly supervised learning based on process-structure-property-performance reciprocity. *Science and Technology of Advanced Materials* **19,** 649–659 (2018).

121. Mysore, S., Kim, E., Strubell, E., Liu, A., Chang, H.-S., Kompella, S., Huang, K., McCallum, A. & Olivetti, E. Automatically extracting action graphs from materials science synthesis procedures. arXiv: `1711.06872 [cs.CL]` (2017).

122. Baevski, A., Edunov, S., Liu, Y., Zettlemoyer, L. & Auli, M. *Cloze-driven Pretraining of Self-attention Networks* in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (Association for Computational Linguistics, Hong Kong, China, 2019), 5360–5369.

123. Li, X., Sun, X., Meng, Y., Liang, J., Wu, F. & Li, J. *Dice Loss for Data-imbalanced NLP Tasks* in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (Association for Computational Linguistics, 2020), 465–476.

124. Kincaid, J. P., Fishburne Jr, R. P., Rogers, R. L. & Chissom, B. S. *Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel* tech. rep. (Institute for Simulation and Training, University of Central Florida, 1975).

125. Kim, J.-D., Ohta, T., Tateisi, Y. & Tsujii, J. GENIA corpus—a semantically annotated corpus for bio-textmining. *Bioinformatics* **19,** i180–i182 (2003).

126. Krallinger, M., Rabal, O., Leitner, F., Vazquez, M., Salgado, D., Lu, Z., Leaman, R., Lu, Y., Ji, D., Lowe, D. M., *et al.* The CHEMDNER corpus of chemicals and drugs and its annotation principles. *Journal of Cheminformatics* **7,** 1–17 (2015).

127. Dieb, T. M., Yoshioka, M., Hara, S. & Newton, M. C. Framework for automatic information extraction from research papers on nanocrystal devices. *Beilstein Journal of Nanotechnology* **6,** 1872–1882 (2015).

128. Kulkarni, C., Xu, W., Ritter, A. & Machiraju, R. An annotated corpus for machine reading of instructions in wet lab protocols. arXiv: `1805.00195 [cs.CL]` (2018).

129. Friedrich, A., Adel, H., Tomazic, F., Hingerl, J., Benteau, R., Maruscyk, A. & Lange, L. *The SOFC-Exp Corpus and Neural Approaches to Information Extraction in the Materials Science Domain* in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (Association for Computational Linguistics, 2020), 1255–1268.

130. Mysore, S., Jensen, Z., Kim, E., Huang, K., Chang, H.-S., Strubell, E., Flanigan, J., McCallum, A. & Olivetti, E. The materials science procedural text corpus: Annotating materials synthesis procedures with shallow semantic structures. arXiv: `1905.06939 [cs.CL]` (2019).

131. Jensen, Z., Kim, E., Kwon, S., Gani, T. Z., Román-Leshkov, Y., Moliner, M., Corma, A. & Olivetti, E. A machine learning approach to zeolite synthesis enabled by automatic literature data extraction. *ACS Central Science* **5,** 892–899 (2019).

132. Milosevic, N., Gregson, C., Hernandez, R. & Nenadic, G. A framework for information extraction from tables in biomedical literature. *International Journal on Document Analysis and Recognition (IJDAR)* **22,** 55–78 (2019).

133. Azimi, S. M., Britz, D., Engstler, M., Fritz, M. & Mücklich, F. Advanced steel microstructural classification by deep learning methods. *Scientific reports* **8,** 1–14 (2018).

134. Matson, T., Farfel, M., Levin, N., Holm, E. & Wang, C. Machine learning and computer vision for the classification of carbon nanotube and nanofiber structures from Transmission Electron Microscopy data. *Microscopy and Microanalysis* **25,** 198–199 (2019).

135. Maksov, A., Dyck, O., Wang, K., Xiao, K., Geohegan, D. B., Sumpter, B. G., Vasudevan, R. K., Jesse, S., Kalinin, S. V. & Ziatdinov, M. Deep learning analysis of defect and phase evolution during electron beam-induced transformations in WS2. *npj Computational Materials* **5,** 12 (2019).

136. Roberts, G., Haile, S. Y., Sainju, R., Edwards, D. J., Hutchinson, B. & Zhu, Y. Deep learning for semantic segmentation of defects in advanced STEM images of steels. *Scientific reports* **9,** 1–12 (2019).

137. Mukaddem, K. T., Beard, E. J., Yildirim, B. & Cole, J. M. ImageDataExtractor: A Tool To Extract and Quantify Data from Microscopy Images. *Journal of Chemical Information and Modeling* **60,** 2492–2509 (2020).

138. Kim, H., Han, J. & Han, T. Y.-J. Machine vision-driven automatic recognition of particle size and morphology in SEM images. *Nanoscale* **12,** 19461–19469 (2020).

139. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. & Wojna, Z. *Rethinking the Inception Architecture for Computer Vision* in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE Computer Society, 2016), 2818–2826.

140. Wasow, T., Perfors, A. & Beaver, D. The puzzle of ambiguity. *Morphology and the web of grammar: Essays in memory of Steven G. Lapointe,* 265–282 (2005).

141. Manning, C. & Schutze, H. *Foundations of statistical natural language processing* (MIT press, 1999).

142. Nickel, M., Murphy, K., Tresp, V. & Gabrilovich, E. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE* **104,** 11–33 (2015).

143. Maas, A., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y. & Potts, C. *Learning word vectors for sentiment analysis* in *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies* (2011), 142–150.

144. Pang, B., Lee, L. & Vaithyanathan, S. Thumbs up? Sentiment classification using machine learning techniques. arXiv: `cs/0205070 [cs.CL]` (2002).

145. Domingos, P. A few useful things to know about machine learning. *Communications of the ACM* **55,** 78–87 (2012).

146. Halevy, A., Norvig, P. & Pereira, F. The unreasonable effectiveness of data. *IEEE Intelligent Systems* **24,** 8–12 (2009).

147. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. & Bengio, Y. Generative adversarial nets. *Advances in neural information processing systems* **27** (2014).

148. Chapelle, O., Scholkopf, B. & Zien, A. Semi-supervised learning. *IEEE Transactions on Neural Networks* **20,** 542–542 (2009).

149. Breiman, L. Random forests. *Machine learning* **45,** 5–32 (2001).

150. McCallum, A. K. Mallet: A machine learning for language toolkit. *http://mallet.cs.umass.edu* (2002).

151. Zhu, S., Fahrenholtz, W. G., Hilmas, G. E. & Zhang, S. C. Pressureless sintering of zirconium diboride using boron carbide and carbon additions. *Journal of the American Ceramic Society* **90,** 3660–3663 (2007).

152. Xiao, X., Chen, L., Wang, X., Li, S., Wang, Q. & Chen, C. Influence of temperature and hydrogen pressure on the hydriding/dehydriding behavior of Ti-doped sodium aluminum hydride. *International journal of hydrogen energy* **32,** 3954–3958 (2007).

153. Liang, C., Wei, M.-C., Tseng, H.-H. & Shu, E.-C. Synthesis and characterization of the acidic properties and pore texture of Al-SBA-15 supports for the canola oil transesterification. *Chemical engineering journal* **223,** 785–794 (2013).

154. Li, G., Zhang, F., Chen, L., Zhang, C., Huang, H. & Li, X. Highly selective hydrodecarbonylation of oleic acid into n-heptadecane over a supported nickel/zinc oxide–alumina catalyst. *ChemCatChem* **7,** 2646–2653 (2015).

155. Zhao, W., Zuo, R. & Fu, J. Temperature-insensitive large electrostrains and electric field induced intermediate phases in (0.7- x) Bi (Mg1/2Ti1/2) O3–xPb (Mg1/3Nb2/3) O3–0.3 PbTiO3 ceramics. *Journal of the European Ceramic Society* **34,** 4235–4245 (2014).

156. Denil, M., Matheson, D. & De Freitas, N. *Narrowing the gap: Random forests in theory and in practice* in *International conference on machine learning* (2014), 665–673.

157. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. & Duchesnay, E. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* **12,** 2825–2830 (2011).

158. Kim, E., Huang, K., Saunders, A., McCallum, A., Ceder, G. & Olivetti, E. Materials synthesis insights from scientific literature via text extraction and machine learning. *Chemistry of Materials* **29,** 9436–9444 (21 2017).

159. Cheng, X., Yan, X., Lan, Y. & Guo, J. Btm: Topic modeling over short texts. *IEEE Transactions on Knowledge and Data Engineering* **26,** 2928–2941 (2014).

160. Xie, P. & Xing, E. P. Integrating document clustering and topic modeling. arXiv: `1309.6874 [cs.LG]` (2013).

161. Yi, X. & Allan, J. *A comparative study of utilizing topic models for information retrieval* in *European conference on information retrieval* (2009), 29–41.

162. Kim, H., Sun, Y., Hockenmaier, J. & Han, J. *Etm: Entity topic models for mining documents associated with entities* in *2012 IEEE 12th International Conference on Data Mining* (2012), 349–358.

163. Guo, H., Zhu, H., Guo, Z., Zhang, X., Wu, X. & Su, Z. *Domain adaptation with latent semantic association for named entity recognition* in *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics* (2009), 281–289.

164. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K. & Dyer, C. *Neural Architectures for Named Entity Recognition* in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (Association for Computational Linguistics, San Diego, California, 2016), 260–270.

165. Prechelt, L. in *Neural Networks: Tricks of the Trade: Second Edition* 53–67 (Springer Berlin Heidelberg, Berlin, Heidelberg, 2012).

166. Řehůřek, R. & Sojka, P. *Software framework for topic modelling with large corpora* English. in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks* (ELRA, Valletta, Malta, May 2010), 45–50.

167. Meurer, A., Smith, C. P., Paprocki, M., Čertík, O., Kirpichev, S. B., Rocklin, M., Kumar, A., Ivanov, S., Moore, J. K., Singh, S., Rathnayake, T., Vig, S., Granger, B. E., Muller, R. P., Bonazzi, F., Gupta, H., Vats, S., Johansson, F., Pedregosa, F., Curry, M. J., Terrel, A. R., Roučka, Š., Saboo, A., Fernando, I., Kulal, S., Cimrman, R. & Scopatz, A. SymPy: symbolic computing in Python. *PeerJ Computer Science* **3,** e103. ISSN: 2376-5992 (2017).

168. Kim, E., Huang, K., Kononova, O., Ceder, G. & Olivetti, E. Distilling a materials synthesis ontology. *Matter* **1,** 8–12 (2019).

169. Young, S. R., Maksov, A., Ziatdinov, M., Cao, Y., Burch, M., Balachandran, J., Li, L., Somnath, S., Patton, R. M., Kalinin, S. V., *et al.* Data mining for better material synthesis: The case of pulsed laser deposition of complex oxides. *Journal of Applied Physics* **123,** 115303 (2018).

170. Davariashtiyani, A., Kadkhodaie, Z. & Kadkhodaei, S. Predicting synthesizability of crystalline materials via deep learning. *Communications Materials* **2,** 1–11 (2021).

171. Sun, W. & Powell-Palm, M. J. Generalized Gibbs' Phase Rule. arXiv: `2105.01337 [math-ph]` (2021).

172. Dinia, A., Vénuat, J., Colis, S. & Pourroy, G. Elaboration and characterization of the Sr2FeMoO6 double perovskite. *Catalysis today* **89,** 297–302 (2004).

173. Shi, T., Xiao, P., Kwon, D.-H., Sai Gautam, G., Chakarawet, K., Kim, H., Bo, S.-H. & Ceder, G. Shear-assisted formation of cation-disordered rocksalt NaMO2 (M= Fe or Mn). *Chemistry of Materials* **30,** 8811–8821 (2018).

174. Yuan, T., Cai, R. & Shao, Z. Different effect of the atmospheres on the phase formation and performance of Li4Ti5O12 prepared from ball-milling-assisted solid-phase reaction with pristine and carbon-precoated TiO2 as starting materials. *The Journal of Physical Chemistry C* **115,** 4943–4952 (2011).

175. Montgomery, D. C., Peck, E. A. & Vining, G. G. *Introduction to linear regression analysis* (John Wiley & Sons, 2021).

176. Friedman, J. H. *The elements of statistical learning: Data mining, inference, and prediction* (springer open, 2017).

177. Azen, R. & Budescu, D. V. The dominance analysis approach for comparing predictors in multiple regression. *Psychological methods* **8,** 129 (2003).

178. Villars, P. & Cenzual, K. *Pearson's Crystal Data: Crystal Structure Database for Inorganic Compounds (on DVD)* Release 2021/22 (ASM International®, Materials Park, Ohio, USA, 2021).

179. Faria, S. & Soromenho, G. Fitting mixtures of linear regressions. *Journal of Statistical Computation and Simulation* **80,** 201–225 (2010).

180. Li, Y. & Liang, Y. *Learning mixtures of linear regressions with nearly optimal complexity* in *Conference On Learning Theory* (2018), 1125–1144.

181. Seabold, S. & Perktold, J. *statsmodels: Econometric and statistical modeling with python* in *9th Python in Science Conference* (2010).

182. Chen, T. & Guestrin, C. *Xgboost: A scalable tree boosting system* in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (2016), 785–794.

183. Quiñonero-Candela, J., Sugiyama, M., Lawrence, N. D. & Schwaighofer, A. *Dataset shift in machine learning* (Mit Press, 2009).

184. Todd, P. K., McDermott, M. J., Rom, C. L., Corrao, A. A., Denney, J. J., Dwaraknath, S. S., Khalifah, P. G., Persson, K. A. & Neilson, J. R. Selectivity in yttrium manganese oxide synthesis via local chemical potentials in hyperdimensional phase space. *Journal of the American Chemical Society* **143,** 15185–15194 (2021).

185. Tammann, G. *Lehrbuch der metallkunde: chemie und physik der metalle und ihrer legierungen* 314 (Leopold Voss, Leipzig, 1932).

186. Merkle, R. & Maier, J. On the tammann–rule. *Zeitschrift für anorganische und allgemeine Chemie* **631,** 1163–1166 (2005).

187. Becker, N. & Dronskowski, R. A first-principles study on new high-pressure metastable polymorphs of MoO2. *Journal of Solid State Chemistry* **237,** 404–410 (2016).

188. Berger, V. W. & Christophi, C. A. Randomization technique, allocation concealment, masking, and susceptibility of trials to selection bias. *Journal of Modern Applied Statistical Methods* **2,** 8 (2003).

189. Cosby, M. R., Mattei, G. S., Wang, Y., Li, Z., Bechtold, N., Chapman, K. W. & Khalifah, P. G. Salt effects on Li-ion exchange kinetics of Na2Mg2P3O9N: Systematic in situ synchrotron diffraction studies. *The Journal of Physical Chemistry C* **124,** 6522–6527 (2020).

190. Chen, S., Hou, Y., Chen, H., Tang, X., Langner, S., Li, N., Stubhan, T., Levchuk, I., Gu, E., Osvet, A., *et al.* Exploring the stability of novel wide bandgap perovskites by a robot based high throughput approach. *Advanced Energy Materials* **8,** 1701543 (2018).

191. Ortiz, B. R., Adamczyk, J. M., Gordiz, K., Braden, T. & Toberer, E. S. Towards the high-throughput synthesis of bulk materials: thermoelectric PbTe–PbSe–SnTe–SnSe alloys. *Molecular Systems Design & Engineering* **4,** 407–420 (2019).

192. Wang, Z., Kononova, O., Cruse, K., He, T., Huo, H., Fei, Y., Zeng, Y., Sun, Y., Cai, Z., Sun, W., *et al.* Dataset of solution-based inorganic materials synthesis recipes extracted from the scientific literature. *Scientific Data* (2022).

193. Cruse, K., Trewartha, A., Lee, S., Wang, Z., Huo, H., He, T., Kononova, O., Jain, A. & Ceder, G. Text-mined dataset of gold nanoparticle synthesis procedures, morphologies, and size entities. *Scientific Data* (2022).

194. Bartel, C. J., Millican, S. L., Deml, A. M., Rumptz, J. R., Tumas, W., Weimer, A. W., Lany, S., Stevanović, V., Musgrave, C. B. & Holder, A. M. Physical descriptor for the Gibbs energy of inorganic crystalline solids and temperature-dependent materials chemistry. *Nature communications* **9,** 1–10 (2018).

195. Bartel, C. J., Trewartha, A., Wang, Q., Dunn, A., Jain, A. & Ceder, G. A critical examination of compound stability predictions from machine-learned formation energies. *npj Computational Materials* **6,** 1–11 (2020).

196. Bartel, C. J., Weimer, A. W., Lany, S., Musgrave, C. B. & Holder, A. M. The role of decomposition reactions in assessing first-principles predictions of solid stability. *npj Computational Materials* **5,** 1–9 (2019).

197. Bartel, C. J. Review of computational approaches to predict the thermodynamic stability of inorganic solids. *Journal of Materials Science,* 1–24 (2022).

198. Kim, S., Chen, J., Cheng, T., Gindulyte, A., He, J., He, S., Li, Q., Shoemaker, B. A., Thiessen, P. A., Yu, B., *et al.* PubChem in 2021: new data content and improved web interfaces. *Nucleic acids research* **49,** D1388–D1395 (2021).

199. Efron, B. Regression and ANOVA with zero-one data: Measures of residual variation. *Journal of the American Statistical Association* **73,** 113–121 (1978).

200. Conneau, A. & Lample, G. Cross-lingual language model pretraining. *Advances in Neural Information Processing Systems* **32,** 7059–7069 (2019).

201. Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H. & Neubig, G. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. arXiv: `2107.13586 [cs.CL]` (2021).

202. Rocktäschel, T., Grefenstette, E., Hermann, K. M., Kočisk, T. & Blunsom, P. Reasoning about entailment with neural attention. arXiv: `1509.06664 [cs.CL]` (2015).

203. Yang, Y., Yih, W.-T. & Meek, C. *Wikiqa: A challenge dataset for open-domain question answering* in *Proceedings of the 2015 conference on empirical methods in natural language processing* (2015), 2013–2018.

204. Moldovan, D., Harabagiu, S., Pasca, M., Mihalcea, R., Girju, R., Goodrum, R. & Rus, V. *The structure and performance of an open-domain question answering system* in *Proceedings of the 38th annual meeting of the Association for Computational Linguistics* (2000), 563–570.

205. Yang, W., Xie, Y., Lin, A., Li, X., Tan, L., Xiong, K., Li, M. & Lin, J. End-to-end open-domain question answering with bertserini. arXiv: `1902.01718 [cs.CL]` (2019).

206. Sutton, R. S. & Barto, A. G. *Reinforcement learning: An introduction* (MIT press, 2018).

207. Nelder, J. A. & Wedderburn, R. W. Generalized linear models. *Journal of the Royal Statistical Society: Series A (General)* **135,** 370–384 (1972).