npj computational materials

www.nature.com/npjcompumats

ARTICLE OPEN

Alloy synthesis and processing by semi-supervised text mining

Weiren Wang[1], Xue Jiang[2,3]✉, Shaohan Tian[1], Pei Liu[1,4,5], Turab Lookman[1,6]✉, Yanjing Su[1]✉ and Jianxin Xie[1]

Alloy synthesis and processing determine the design of alloys with desired microstructure and properties. However, using data science to identify optimal synthesis-design routes from a specified set of starting materials has been limited by large-scale data acquisition. Text mining has made it possible to convert scientific text into structured data collections. Still, the complexity, diversity, and flexibility of synthesis and processing expressions, and the lack of annotated corpora with a gold standard severely hinder accurate and efficient extraction. Here we introduce a semi-supervised text mining method to extract the parameters corresponding to the sequence of actions of synthesis and processing. We automatically extract a total of 9853 superalloy synthesis and processing actions with chemical compositions from a corpus of 16,604 superalloy articles published up to 2022. These have then been used to capture an explicitly expressed synthesis factor for predicting γ′ phase coarsening. The synthesis factor derived from text mining significantly improves the performance of the data-driven γ′ size prediction model. The method thus complements the use of data-driven approaches in the search for relationships between synthesis and structures.

npj Computational Materials (2023)9:183 ; https://doi.org/10.1038/s41524-023-01138-w

INTRODUCTION

The discovery of materials with targeted properties requires a seamless, integrated approach combining experiments, theory, and computations. The paradigm of Composition-Synthesis/Processing-Structure-Property-Performance[1,2] often serves as a guide to exploration towards this task. The challenge in discovery is that the large materials space consists of innumerable combinations of components and structures, which is strongly determined by the potentially synthesized route[1]. Over the last few years, machine learning (ML) has guided the search for new materials using a data-driven approach[3–8]. For example, materials synthesis is beginning to see dramatic improvements in efficiencies due to the integration of ML capabilities and robotic control of synthetic planning and automated experiments for flow reactors, photovoltaic films, organic synthesis and perovskites by mobile robotics[9–14]. However, the design of alloys with desired properties requires not only dealing with increasing chemical and structural complexity, but also a myriad of processing routes. The materials space is just too vast for today's synthesis capabilities. Hence, our focus here is to use data science to identify optimal synthesis-design routes to produce a desired alloy from a specified set of starting materials[15,16].

Early approaches towards materials data extraction from scientific articles have essentially been manual in nature[17]. The dramatic development of text mining and natural language processing (NLP) techniques have made it possible to convert scientific text into ML-oriented data collections[18–20]. Recently, NLP pipelines for automatic data extraction from journal articles of chemical composition and properties of organic and inorganic chemical compounds, as well as super and aluminum alloys have been introduced[19,21–25]. Alloy synthesis and processing information are usually described in the form of continuous events, and the actions are sequentially dependent. There are various types of actions, flexible expressions, and different conditions and parameters, and meanwhile, continuous synthesis and processing events are often mixed with a large number of discussions on experimental phenomena and intermediate products, which bring great challenges to actions and parameters extraction. Nevertheless, mature, deep learning (DL) provides powerful capabilities to analyze unstructured data and identify features automatically. The well-documented libraries make use of DL more accessible. Kim et al. labeled 20 articles (~5200 words) on oxide materials and trained a neural network to recognize synthesis parameters with an F1 score of 81%[18]. Kononova et al. manually annotated the operation entities from 834 solid-state synthesis paragraphs of 750 papers and trained a bidirectional long short-term memory (BiLSTM) network with a conditional random field (CRF) layer (BiLSTM-CRF) model with an F1 score of 90%[26]. Huo et al. designed a qualitative topic extraction method related to experimental protocols rather than recognizing detailed processing parameters. They clustered the sentences into topics and then trained a classification model to predict the latent topics of unseen experimental sentences with an F1 score above 90%[27]. Despite numerous advantages, a DL model uses a few thousand to millions of parameters. To train a DL-based NLP named entity recognition (NER) and information extraction (IE) model requires many high-quality annotations. For alloys with structures and properties strongly determined by synthesis and processing routes, the limited amount of corpus and lack of high-quality annotations severely hinder accurate and efficient extraction.

As a core material for the most advanced aero engines and industrial gas turbines, the synthesis and processing of superalloys impacts their design with desired microstructure and properties[28–31]. We previously introduced an NLP pipeline to capture both chemical compositions and property data from 14425 articles published before 2020 on superalloys[25]. A rule-based NER method and a distance-based heuristic IE were proposed to overcome the drawback of a limited set of labeled corpora guaranteeing high precision and recall simultaneously. Under such conditions, the

[1]Beijing Advanced Innovation Center for Materials Genome Engineering, Institute for Advanced Materials and Technology, University of Science and Technology Beijing, 100083 Beijing, China. [2]Collaborative Innovation Center of Steel Technology, University of Science and Technology Beijing, 100083 Beijing, China. [3]Liaoning Academy of Materials, Shenyang 110000 Liaoning, China. [4]Beijing Key Laboratory of Advanced High Temperature Materials, Central Iron & Steel Research Institute, 100081 Beijing, China. [5]Beijing GAONA Materials & Technology Co., LTD, 100081 Beijing, China. [6]AiMaterials Research LLC, Santa Fe, NM 87501, USA. ✉email: jiangxue@ustb.edu.cn; turablookman@gmail.com; yjsu@ustb.edu.cn

rule-based method is efficient compared to a DL model due to the relatively low diversity of entity categories which can be handled well by human expertise. However, the entity categories and their relationships to synthesis and processing information are more complex and flexible in formation so that a rule-based method can become cumbersome and expensive. Thus, supervised DL typically requires labeling a large but expensive corpus, as well relabeling the corpus when IE is oriented to a new field. The rule-based strategies require undue human intervention to get started. We therefore employ semi-supervised intuition in this work to leverage a relatively small amount of labeled and large amount of unlabeled data to bolster model performance.

We introduce a semi-supervised text mining method to extract the parameters corresponding to the sequence of actions of synthesis and processing from a corpus. This makes it possible with less domain-specific experience and corpus annotation to achieve relatively high IE performance for superalloy synthesis and processing, that is, we extract details of the synthesis process. A semi-supervised recommendation algorithm for token-level action and a multi-level bootstrapping algorithm for chunk-level actions are developed for a small corpus with few annotations so that a small number of seeds are required to initiate the learning process. The F1 score of action entity recognition reaches 89.28%, much higher than the 74.95% achieved via the BiLSTM-CRF model. In total, 9,853 superalloy synthesis and processing actions with chemical compositions are automatically extracted from a corpus of 16,604 superalloy articles from Elsevier and other publishers.

To evaluate the accuracy and diversity of the extracted results, we visualized the data from multiple perspectives to distill scientific insights. We analyzed superalloy synthesis processes to determine which are of wide current interest, and we show how temperatures for solution and aging treatments are correlated. We also determined the transition probabilities from one action to another in a given synthesis process. A superalloy synthesis factor combining solution temperature (S_c), aging temperature (A_c) and aging time (A_t), in form of $(A_t{*}S_c{}^{0.5})^{0.5} + A_c$, is inferred by symbolic regression (SR), illustrating a positive correlation with γ′ phase coarsening. This synthesis factor derived from text mining significantly improves the performance of the data-driven γ′ size prediction model on the superalloys reported subsequently in 2023 and which we synthesized. Thus, semi-supervised text mining enables us to complement data-driven approaches for understanding relationships between synthesis and structures.

RESULTS

Extraction strategy

The core stages of our automated text mining pipeline for superalloys synthesis and processing involves action dictionary generation, NER, and dependency parsing, in addition to several necessary NLP stages such as article retrieval and preprocessing, paragraph classification, table parsing, and interdependency resolution. The schematic overview of the synthesis extraction is shown in Fig. 1. For scientific article retrieval and preprocessing, the raw archived corpus was parsed and organized in paragraphs. After paragraph classification, the paragraphs related to the concrete synthesis procedures were automatically selected. Action dictionary generation can generate token-level and chunk-level synthesis actions semi-automatically. NER methods are designed to recognize the action entities based on the generated dictionary. Dependency parsing establishes specific tuple relationships for actions and parameters in terms of latent semantics, and interdependency resolution resolves the linkage between chemical composition of mentioned samples and their actions in the specific synthesis process. Finally, the extracted superalloy synthesis information containing the article digital object identifier (DOI), sample composition, synthesis action sequence with parameters is automatically compiled into a structured (Comma-Separated Values, CSV) and semi-structured (JavaScript Object Notation, JSON) format to form a materials database for reuse.

Semi-supervised action dictionary generation

It is usual for target entities to be recognized by DL models trained on large corpora with hundreds of thousands of tokens. This requires accurate labels for each category of entity and appropriate annotation strategy for different types of corpus[26,32]. There are thousands of synthesis and processing superalloy actions discussed in the superalloy literature, although the number of articles is only about 16,000. The synthesis and processing actions in superalloy corpora are described in token-level and chunk-level entities depending on the phrase length, in contrast to chemical synthesis where mainly token-level action entities are involved. Moreover, according to the position of action in the superalloy process routine, the description for the same action is also different, for example, aging treatment may exist in diverse forms such as primary aging, secondary aging, etc. This introduces further challenges in the manual fine labeling of entities. We therefore propose a semi-supervised method to allow for a rapid startup by generating a complete process action dictionary based on the literature corpus for further action entity recognition, which

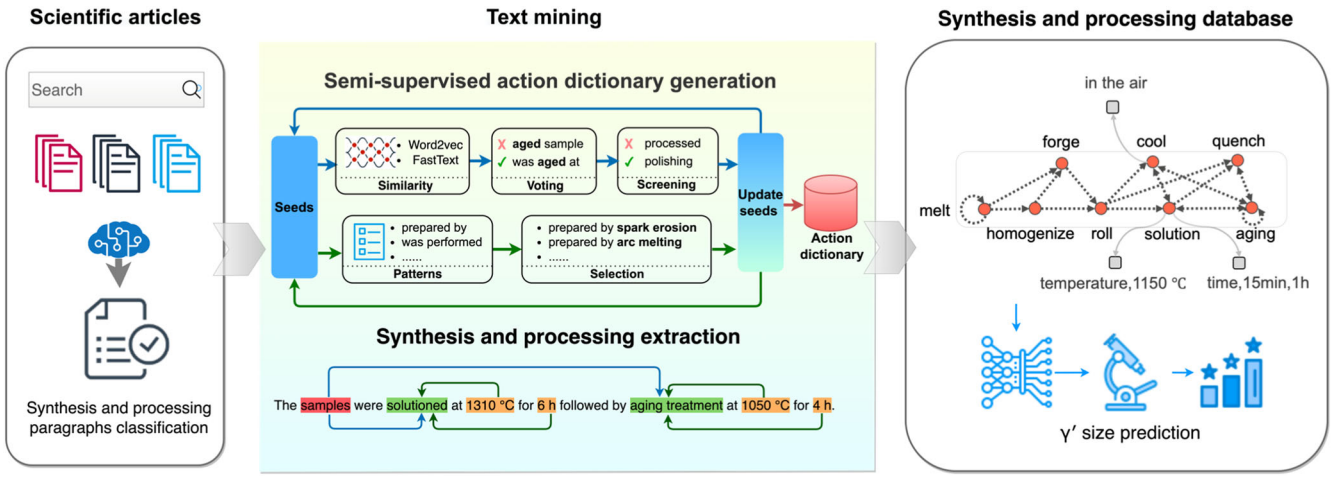

Fig. 1 The schematic overview of synthesis extraction.

Table 1. Examples for the parsing grammar of token-level and chunk-level action entities.

Category	Entity POS	Syntactic templates	Examples
Token-level	VB	VB[DNP]\sTO <action>	… and allowed to **soak** for …
	VBD/VBN/JJ	VB[DNP][\sRB]* <action>	… were mechanically **ground** …
	VBG	VB[DN]\sIN <action>	… followed by **aging** at …
Chunk-level	NP/VP	NP\sVB[GZ] <action>	… sample was **solution treated** …
	NP/VP	VB[DGN]\sIN <action>	… prepared by **vacuum induction melting** …
	NP	VB[DGN]\sTO <action>	… subject to a **solution heat treatment** …
	NP/VP	IN\sNN\sIN <action>	… by method of **directional solidification** …
	NP	<action> VBD\sVBN	… **solution heat treatment** was performed on …

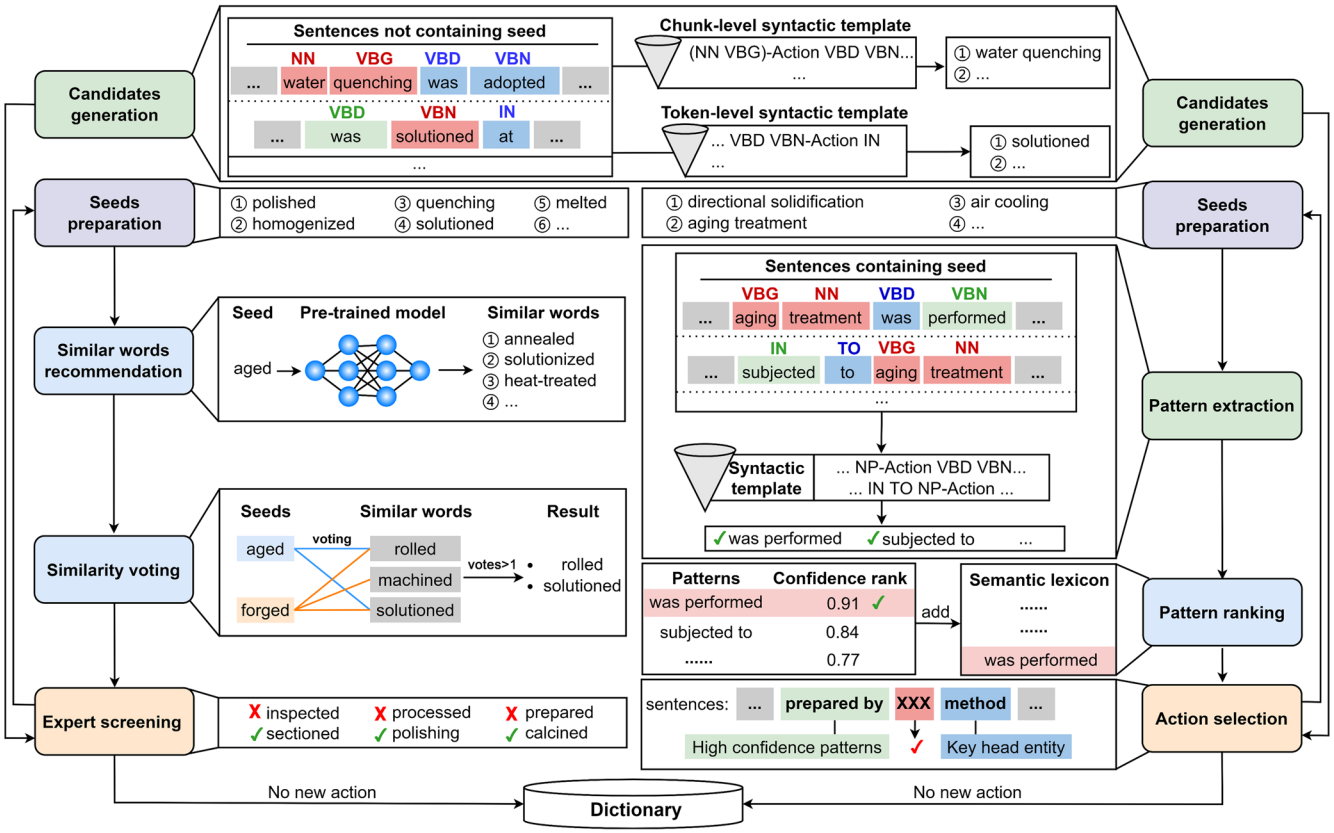

Fig. 2 Schematic workflow of the semi-supervised action dictionary generation method. The left flowchart shows the token-level action recommendation algorithm, and the right shows the multi-level bootstrapping algorithm for chunk-level actions. Eventually the generated token- and chunk-level entities are merged into the action dictionary. Tags shown: NN = noun, CD = cardinal number, VBG = verb (present participle or gerund), VBD = verb (past tense), VBN = verb (past participle), IN = preposition, JJ = adjective, TO (preposition or infinitive marker).

only needs to provide a small number of seed actions based on experience to go with the few rounds of manual checks. Table 1 gives some examples for token-level and chunk-level action entities with their Part-of-Speech (POS) and syntactic templates. Token-level action entities mainly involve nouns and verbs, whereas chunk-level entities belong to noun and verb phrases.

The proposed semi-supervised action dictionary generation method is shown in Fig. 2. The token-level entity recommendation algorithm (left) involves four stages of candidate generation, seed preparation, similar words recommendation, similarity voting and expert screening. The multi-level bootstrapping algorithm for chunk-level action (right) comprises the stages of candidate generation, seed preparation, pattern extraction, pattern ranking and action selection.

Token-level action recommendation algorithm

Candidate entities, starting seeds, and two pre-trained embedding models are required before execution of the token-level action recommendation algorithm. The candidate entities can provide the largest collection of potential actions according to the POS of the token entity, the syntactic template of its pre-context and post-context from all positive paragraphs, illustrated in Table 1. As for starting seeds, the more seeds provided initially, the fewer iterations of the algorithm, but the overall generated dictionary size remains the same. Here only 20 seeds are provided by experts to start our algorithm. The token-level starting seeds are further used to obtain similar words based on the word embedding models Word2Vec and FastText. These were initially pre-trained on approximately 16,000 unlabeled full-text superalloy articles and

used for calculating the cosine similarity between the candidate entity and starting seed. Word2vec helps find the most similar syntactic and semantic words, and FastText constructs word embeddings from character-level n-gram representations to find words with greater morphological similarity[33–36]. The details of word embedding models can be found in Method. Following the recommendation of similar entities, a voting process is performed to obtain action entities with high confidence. In this process, a seed is considered to have voted for the entity if the similarity by Word2vec between the candidate entity and the seed is higher than the threshold *w*, or if the similarity by FastText between the candidate entity and seed is higher than *fw*. The thresholds *w* and *fw* need to be optimized during the application of the algorithm. This affects the recommended precision and number of new generated actions in each iteration. Thus, these two parameters are adjusted to trade-off a balance. The parameter optimization for *w* and *fw* in the semi-supervised token-level action entity recommendation algorithm can be found in Supplementary Fig. 1. As *w* and *fw* increase, the size of the generated dictionary gets smaller, and the recommended precision gets higher. When *w* is equal to 0.46, there are 697 recommended tokens with little manual intervention. When *fw* is equal to 0.7, the recommended number and precision are both high.

Entities that receive more than 2 votes and belong to a candidate is selected and passed to the expert for manual screening. The entities screened out by the expert are added back to the seeds and participate in the next iteration. If there is no new action entity generated and passed to the expert, the iteration will end. The pseudocode of the token-level actions generated algorithm is shown in Supplementary Fig. 2. The final generated dictionary for token-level actions contains 717 action entities (including 20 starting seeds and recommended actions) and is used for the subsequent action entity recognition.

The multi-level bootstrapping algorithm for chunk-level actions

The original multi-level bootstrapping algorithm was published by Ellen to generate a dictionary for noun phrases, such as the names of people, companies, or locations[37,38]. We first extend this multi-level bootstrapping algorithm to generate suitable noun and verb phrases for synthesis and processing action entities. Moreover, the original algorithm assigns scores to entities according to the type of pattern, not taking the extraction frequency of each entity into consideration, so that it is hard to distinguish different entities obtained from the same pattern. We improve this multi-level bootstrapping algorithm to generate both the collection of chunk-level action entities and patterns simultaneously and optimize the entity confidence score method. The modified multi-level bootstrapping algorithm for chunk-level actions (Fig. 2 right workflow) also starts with candidate generation and seed preparation. The candidate chunk-level entities, which are generated by the POS of chunk-level actions and the syntactic template of its pre-context and post-context in Table 1, can provide the largest collection of potential actions in the action selection stage. The starting seeds are provided by experts and contain both noun and verb phrases, here 20 seeds picked from 5 articles were manually provided.

A pattern is the word sequence in front of or behind the seed entity, and it is the most important contextual feature for chunk-level action generation. Pattern extraction is designed to find all patterns that have a co-occurrence constraint with seeds and, also consistent with the syntactic template. The corresponding sentence that contains the seed is divided into three parts: "before", the seed, and "after". The "before" is a sequence of tokens located in front of the seed entity and the "after" is a sequence of tokens behind the seed. The parameter of window_size is used to control the "before" and "after" size and depends on the length of the syntactic templates. For example, the window_size of syntactic template IN\sNN\sIN is 3 and that of VB[DNP]\sTO is 2. If the POS of a "before" or "after" matches any customized syntactic template, the corresponding "before" or "after" sequence will be added to the set of patterns. The patterns are obtained when all the sentences that contain seeds are traversed during an iteration.

Pattern ranking recommends the best pattern amongst all patterns after pattern extraction. During each iteration, the confidence associated with the pattern is calculated with Eq. (1) and Eq. (2)[37,38], where $Pattern_i$ stands for the i-th pattern, F_i is the number of unique seeds hit by $Pattern_i$, N_i is the total number of unique chunk-level entities that $Pattern_i$ can extract, R_i represents the precision (probability) of the pattern to extract relevant information and $\mathrm{Score}(\mathrm{Pattern_i})$ balances the reliability (R_i) and frequency (F_i).

$$R_i = F_i/N_i \tag{1}$$

$$\mathrm{Score}(Pattern_i) = R_i * \log_2 F_i \tag{2}$$

The best pattern with the highest confidence score will then be added to the semantic lexicon. The Supplementary Table 1 shows the best patterns in the first 6 iterations of mutual bootstrapping.

After pattern ranking, the patterns in the semantic lexicon are then used to select optimal chunk-level actions. From a probability perspective, an entity that is hit by more patterns will be more likely to be an action entity. Therefore, anther confidence score, $\mathrm{Score}(Entity_i)$, is defined to reflect the frequentist likelihood that a candidate chunk-level entity is extracted by patterns in the semantic lexicon, shown in below equation.

$$\mathrm{Score}(Entity_i) = \sum_{k=1}^{N_i}(1 + \mathrm{Score}(Pattern_k) + k * Count_{seed}) \tag{3}$$

where N_i is the number of patterns in the semantic lexicon that can successfully extract the entity. For each pattern, $Count_{seed}$ is the number of seeds that can match the pattern. The k is the weight of $Count_{seed}$ and affects the number and precision of recommended entities in each iteration. This also requires to be optimized to get a better token dictionary with higher accuracy and a larger size. The $\mathrm{Score}(Pattern_k)$ can be calculated using Eq. (2). The entity with the highest score is then likely to be selected.

Whether the entity with the highest score can eventually be added back to the seeds depends on the following constraints: a minimum confidence threshold, T_c, for action selection, with the entity satisfying the criteria score $\geq T_c$, and the new entity is lemmatized and added back to the seeds for next iteration. If the highest $\mathrm{Score}(Entity_i)$ in an iteration falls below T_c, the iteration ends. The parameter optimization for T_c and k in the multi-level bootstrapping algorithm can be found in Supplementary Fig. 3. When $T_c = 2$ *and* $k = 1$, the recommended number and precision of chunk-level actions are both high. The pseudocode of the multi-level bootstrapping algorithm is shown in Supplementary Fig. 4.

The above semi-supervised recommendation algorithm and multi-level bootstrapping algorithm for token-level and chunk-level actions generation were applied to a total of 14487 target paragraphs classified from approximately 16,000 articles, and 697 new token-level action entities (except for the 20 initial seeds) and the 1199 chunk-level action captured entities were compiled into the action dictionary.

Named entity recognition

The above generated action dictionary can be further used for NER of the synthesis and processing actions in the superalloy corpus. Fig. 3 shows the action NER workflow by POS tagging, POS screening, and relaxed matching. The input sentence is parsed by POS tagging to identify all verbs, NPs, VPs, and their contexts, and the tagging results are the input into the POS screening to

recognize the entities that meet the contextual POS rules and have a similar context with actions in dictionary, as demonstrated during semi-supervised action dictionary generation. As the same action is usually expressed in various formats, such as "aged", "aging treatment", and "two-step aging", a relaxed match strategy is particularly useful in recognizing more latent entities during NER. If the entity after POS screening matches part of the chunk-level actions or comprises the token-level actions in the generated dictionary, it will be recognized as an action entity.

In addition to action entities, the synthesis parameters with units are also important for synthesis and processing. We use regular expressions to recognize these entities as reported previously[25]. We also compared our action entity extraction method with the BiLSTM-CRF model. The action NER was validated on 348 sentences (approximately 8800 words) randomly selected from 1308 sentences used in the BiLSTM-CRF model. The obtained precision, recall, and F1 score are listed in Table 2. Compared the BiLSTM-CRF model, our proposed semi-supervised method performs better, including the token-level and chunk-level entities. The details of BiLSTM-CRF and its results can be found in Method and Supplementary Table 2.

Dependency parsing

Dependency parsing aims to solve linkage between the action entity and its parameters. Here we infer the structural and semantic relation for each action entity and construct the parsing tree based on the dependence grammar[39]. The edge with a tag in the parsing tree represents the dependent relation between the starting point entity and its subordinate entity. The tag in the directed edge represents the syntactic role in the dependent relations. As shown in Fig. 4, after NER, the sentence can be parsed into subject, action, and parameter entities. According to the

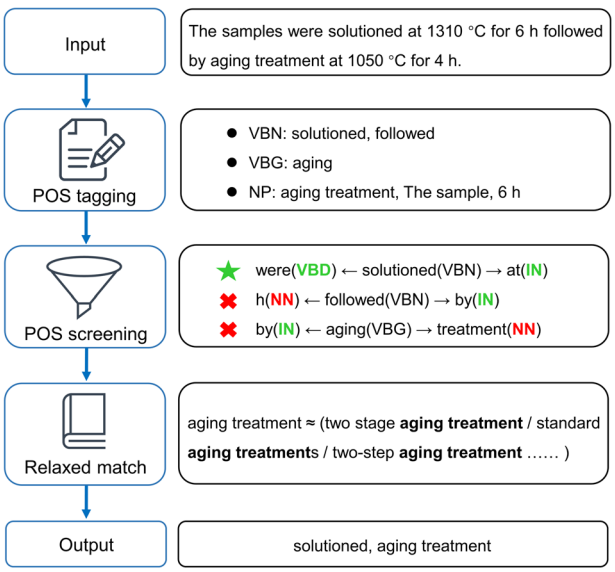

Fig. 3 NER for action entities by POS tagging, POS screening, and relaxed match.

entity category and its POS, the original action and parameter entity need to be replaced in a more normalized format to help capture the structural and semantic relations accurately. The replacement follows the rules: for entities with POS of VBN or VBD, the entity is replaced by the form "id"+"Ved" such as "1Ved". The "id" refers to the order in which it appears in the sentence. The VBG entity is replaced by "id"+"Ving", such as "2Ving". For the NP entity, the format is "id"+"NP", *i.e.* "1NP".

After preprocessing, the sentence is parsed to obtain a dependency parse tree and three-tuple relations among entities using the Stanford CoreNLP package[40]. Amongst all types of relations, Nsubj is defined as the relation from a subject to a verb, which represents the relation from sample to action when the verb belongs to the action entity. Obl is defined as the relation from an object to a verb, which represents the relation from parameter to action when the verb belongs to the action entity and the object belongs to the parameter entity. This matching rule is used to interpret the dependency parsing results of the sentence and yield three-tuples with the target entity.

In total, we have captured 55206 actions from 16,604 superalloy articles. There are 13,211 actions that can be related to concrete synthesis parameters. The precision, recall, and F1 score for actions are shown in Table 3, which were manually validated on 30 randomly sampled articles. For each paragraph, the captured action tuples with parameters can be linked as action sequence, such as <arc melted→homogenized→rolled→solutioned (1250 °C, 5 h)→air cool→aged (1000 °C, 3 h)→air cool > .

The concrete chemical composition information for each sample was obtained from tables by our SuperalloyDigger pipeline[25]. From the ~16,000 articles, we automatically extracted a total of 20,368 chemical composition instances. We performed interdependency resolution to map the composition of samples with the synthesis and processing routine (see Interdependency resolution in "Method"). In total, we merged 9853 complete records with composition and synthesis actions with parameters from 20,368 composition instances and 13,211 instances with actions and parameters.

DISCUSSION

The proposed semi-supervised extraction and tree-based dependency parsing capture synthesis and processing information of superalloys by overcoming the drawbacks of limited corpus labels. We now evaluate the extracted results, including visualize the data to glean scientific insights.

We first visualize the coverage and diversity of the extraction results from the perspective of time and action category. The heatmap (Fig. 5) represents the frequency of various actions such as quenching, aging, cutting, solution treatment and cooling as a function of year from 2004 to 2021. For each category (the row in Fig. 5), there are multiple subdivisions of synthesis actions such as "investment casting", "ingot casting" and "single crystal casting" in casting category. Our semi-supervised text mining method can capture expressions corresponding to such diverse action information. We can also see there has been increasing activity in superalloy technologies since 2013 and greater emphasis in time on actions employing quenching, aging, cutting, solution

Table 2. Precision, recall, and F1 score of the action NER by our work and BiLSTM-CRF.

Category		Precision	Recall	F1 score	Validated on
Action	Our work	90.58%	88.03%	89.28%	348 sentences
	BiLSTM-CRF	77.02%	72.98%	74.95%	1308 sentences (fivefold cross validation)
Parameter (with unit)	Our work	98.49%	94.91%	96.67%	348 sentences
	BiLSTM-CRF	94.11%	80.85%	86.98%	1308 sentences (fivefold cross validation)

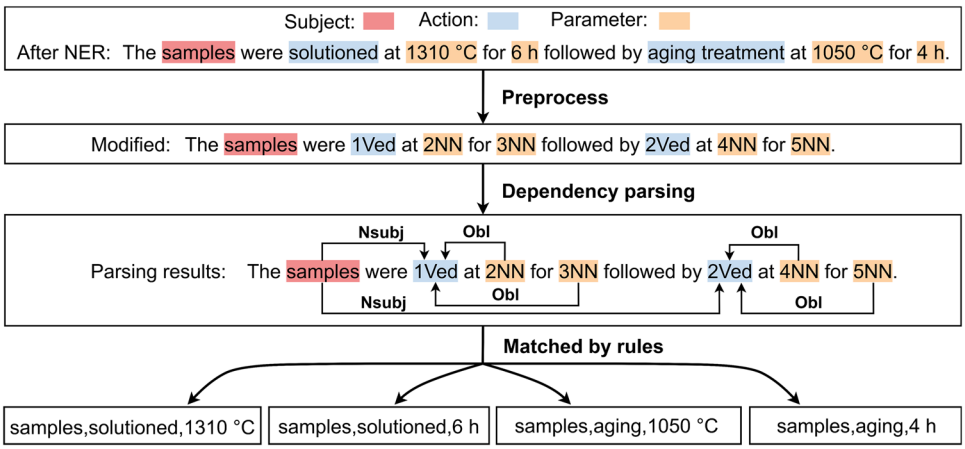

Fig. 4 Schematic overview of dependency parsing process.

Table 3. Precision, recall, and F1 scores of token-/chunk- level actions and parameters.

Category	Extracted instance number from all superalloy articles	Precision	Recall	F1 score	Validated on
Token-level action	25,243	92.85%	90.28%	91.55%	30 articles (348 sentences)
Chunk-level action	29,963	88.23%	85.71%	86.95%	30 articles (348 sentences)
Action + parameter	13,211	81.31%	84.09%	82.68%	30 articles (348 sentences)
Action in total	55,206	90.58%	88.03%	89.28%	30 articles (348 sentences)

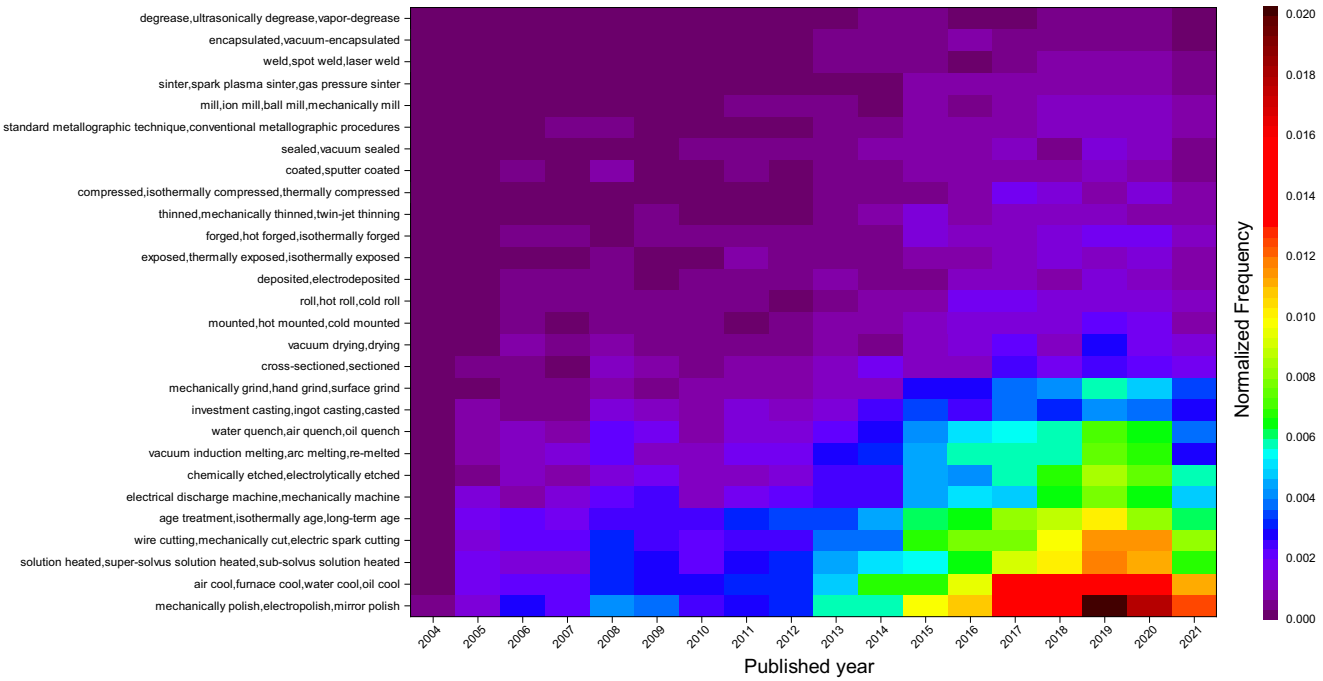

Fig. 5 A heatmap of the frequency of actions reported from 2004 to 2021.

treatment, cooling and polishing. Heat treatments impact phase morphology and mechanical properties of superalloys, and the choice of appropriate treatment is still widely studied as it governs the size and shape of microstructures and properties such as strength, hardness, and ductility.

Temperature is critical to ensure the fine melt of precipitated phases and precipitation of strengthening phases in synthesis. Fig. 6 depicts temperatures during solution and aging treatments for various superalloys. In Fig. 6a, the extracted dataset that contains both the solution temperature and compositions is split into two classes based on the relative content of Ta, Ti, and Cr. High Cr, low Ti and Ta represents Cr higher than 6%, with the total of Ta and Ti not lower than 3%. Low Cr, high Ti and Ta represents Cr lower than 6% with the total of Ta and Ti higher than 3%. The superalloys with low Cr, high Ti and Ta have solution temperatures generally higher than those with high Cr, low Ti and Ta. This is because the γ′ solvus

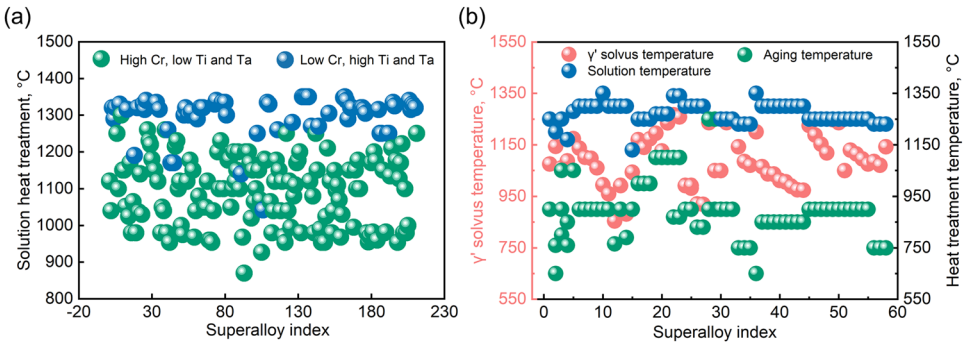

Fig. 6 Solution and aging treatment temperatures for various superalloys. a The solution temperatures dependent on the composition of Cr, Ti and Ta. **b** The γ′ solvus, aging treatment, and solution treatment temperatures for each alloy.

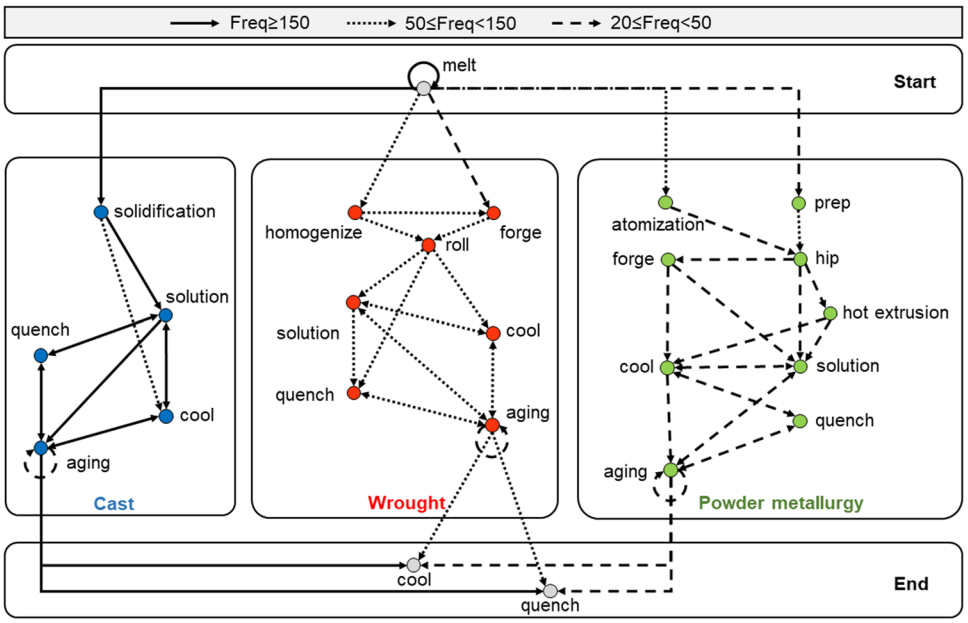

Fig. 7 The flowchart of transition probabilities from one action to another for cast, wrought and power metallurgy superalloys.

temperature is significantly increased with the addition of Ta and Ti but decreases with Cr, so the alloys with high Ti and Ta need higher solution temperatures (above 1250 °C), consistent with results reported by Chen[41,42]. Coupled with our previous extracted γ′ solvus temperature dataset[25], Fig. 6b shows the relationship between γ′ solvus, aging treatment and solution treatment temperatures. The γ′ solvus temperature for each alloy lies between its solution and aging temperatures, in agreement with known rules[43].

In materials synthesis, experimental steps usually follow a certain order specific to the synthesis methodologies. We linked the extracted actions in order to obtain an action sequence for each article. By calculating the transition probability from one action to another in one action sequence, a Markov chain representation to show how various experimental steps proceed was constructed (Fig. 7). In the flowchart, the directed graph consists of nodes and directed edges, and a node represents an experimental action, and an edge represents a transition from one action to another. The solid black line indicates that the transfer from one action to another occurs at a frequency≥150, and two dashed lines represent 50≤frequency <150 and 20≤frequency <50. The high transfer frequency means that the possibility of inferring the latter action from the previous action is greater in synthesis.

There are in total three types of synthesis processes shown in Fig. 7, including casting, wrought and powder metallurgy. The extracted data also contains insights with adjacent relations as the sequence in casting: “solidification”-> “solution”-> “aging”-> “quench” -> “solution” -> “aging” etc., which matches expert intuition. Also, for powder metallurgy, actions “cool” and “quench” usually follow “solution” and “aging”, but “aging” never follows “hot extrusion”. The bidirectional edges are found between “cool”, “solution” and “aging”, indicating it is a common practice to repeat “solution” and “aging” in synthesis / processing steps. The constructed Markov chain in Fig. 7 captures experimental steps for different synthesis processes, indicating confidence in the extraction procedures.

The coarsening of γ′ precipitates of superalloys is greatly influenced by several factors, such as composition, solution, and aging procedure. Here we further utilized the extracted synthesis and processing dataset to predict the coarsening of γ′ precipitates to reveal a critical synthesis factor. The size of γ′ precipitates was used to depict the coarsening behavior automatically captured by our SuperalloyDigger NLP pipeline[25]. After data preprocessing, a high-quality dataset with 137 records was obtained containing the γ′ size, composition space of Co, Al, W, Ni, Ti, Cr, Ta, Mo, Re, and Nb, and synthesis conditions of solution temperature, solution time, aging temperature, and aging time. SR was then adopted to capture the explicitly expressed synthesis factor for γ′ size by genetic programming SR (GPSR) implemented in the gplearn code[44].

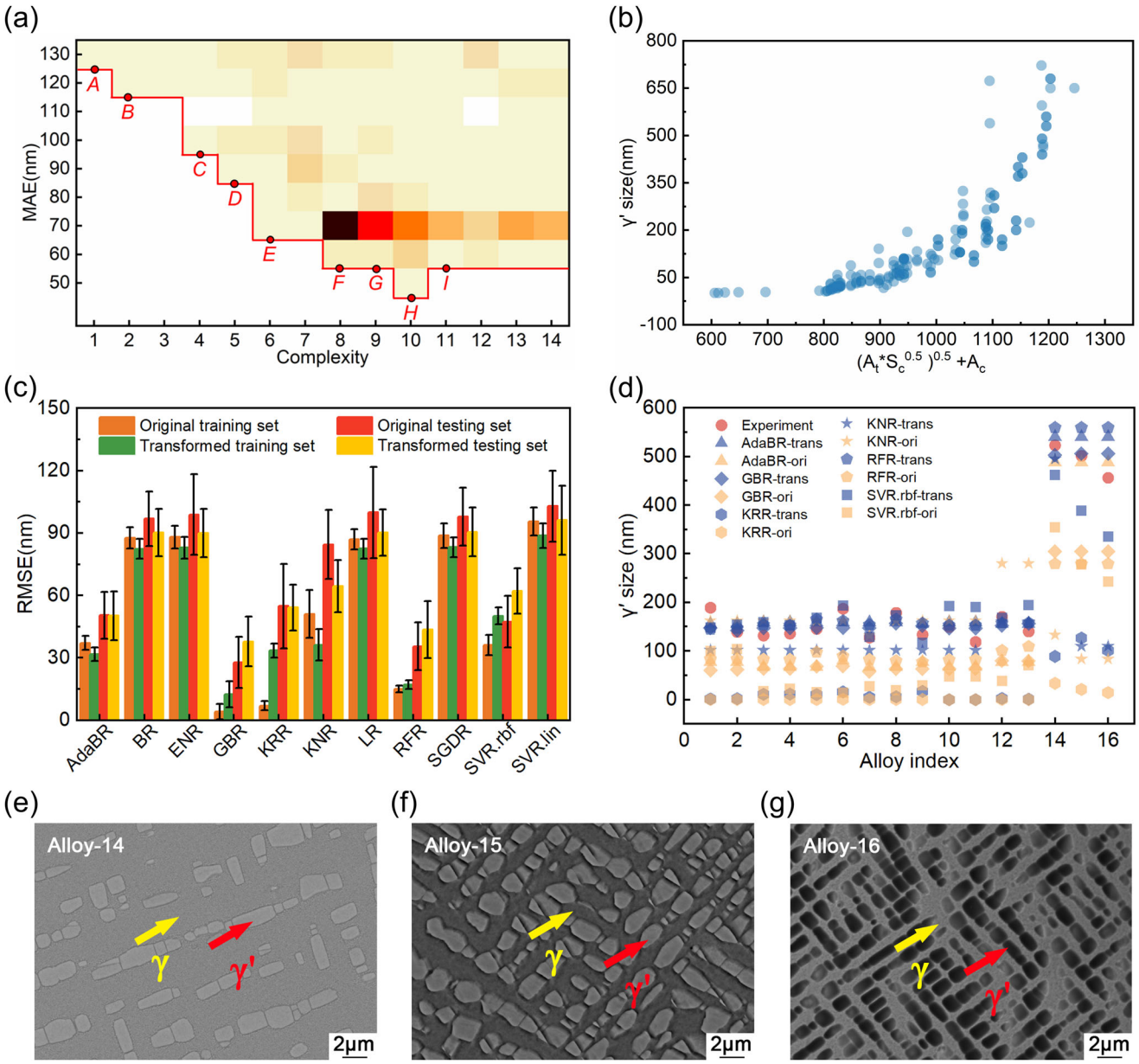

Fig. 8 The generated superalloy synthesis factor by SR which greatly improves γ′ size prediction performance. a Pareto front of MAE vs. complexity among 60,000 mathematical formulas shown via density plot. **b** Scatter plot of γ′ size vs $(A_t{*}S_c^{0.5})^{0.5} + A_c$. **c** RMSE for model selection under original and transformed feature space by fivefold cross validation. **d** The measured and predicted γ′ size of 13 superalloys newly reported in 2023 and 3 superalloys which we synthesized among all models. **e** The microstructure for alloy Co-29.6Ni-10.8Al-2Ti-2.5W-1.6Ta-1Mo-3.5Cr. **f** The microstructure for alloy Co-30Ni-10.4Al-1.5Ti-1.6W-3Ta-1Mo-4.9Cr. **g** The microstructure for alloy Co-29.9Ni-10.4Al-1.9Ti-1W-3.3Ta-1.1Mo-5.2Cr-0.8Re.

(The details of SR can be found in Methods). Here the complexity and mean absolute errors (MAE) are used for the metrics for produced formulae from SR, and the complexity refers to the number of arithmetic operators, including addition, subtraction, division, multiplication, and square root. A total of 60,000 candidate formulae were generated and sorted using MAE under the same complexities, shown in Fig. 8a. There are 9 mathematical formulae (marked A–I in Fig. 8a) located at the Pareto front with low complexity and MAE. The specific formulae are shown in Supplementary Table 3. Although the accuracies of these formulas are not high enough owing to the white box modeling, we can infer that A_t, S_c, A_c and the term $(A_t{*}S_c^{0.5})^{0.5}$ occur frequently and therefore appear to play a significant role in determining γ′ size. To couple with these three synthesis parameters, certain terms, such as $(A_t{*}S_c^{0.5})^{0.5} + A_c$, $(A_t{*}S_c^{0.5})^{0.5} + A_c^{0.5}$ and $(A_c{*}A_t)^{0.5} + S_c^{0.5}$ were considered in order to study their relationship with γ′ size through a scatter plot. In particular, $(A_t{*}S_c^{0.5})^{0.5} + A_c$ shows a positive correlation, *i.e.*, exponential growth, with γ′ size shown in Fig. 8b.

The study of precipitate evolution is important for materials design. Classical physical models predict the ripening behavior of particles. The LSW theory assumes a very dilute environment without interactions among particles to predict the ripening behavior of γ′ precipitates[45]. Ardell incorporated the influence of finite precipitate volume fraction into the framework of diffusion-controlled coarsening kinetics and modified LSW (MLSW). From classical kinetic theory, γ′ size coarsening without coalescence is

predicted to obey Eq. (4)[46–48]:

$$<r_t>^3 - <r_0>^3 = \frac{8D_0 \exp\left(\frac{-Q}{RT}\right)\sigma C_i^{\gamma}\left(1 - C_i^{\gamma}\right)V_m}{9RT\left(C_i^{\gamma'} - C_i^{\gamma}\right)^2}(t - t_0) \quad (4)$$

As shown in Eq. (4), T is the aging temperature and t is the aging time, but the equation is established under ideal conditions (all phases are dissolved under the solution temperature). In addition, an elevated S_c decreases the residual dendritic segregation of refractory elements (i.e., Re and W) and suppresses the precipitation of deleterious topologically close-packed (TCP) phases in Ni-based superalloys. Unsuitable S_c promotes the formation of TCP, which reduces the concentration of certain solid solution strengthening elements such as Cr, Mo, W and Re in the γ phase. The lower element concentrations will also lead to a lower rate for the coarsening of γ′ phase. Thus, S_c impacts the coarsening of γ′ phase.

To test this factor, we constructed an ML-based γ′ size prediction model with $(A_t{*}S_c{}^{0.5})^{0.5} + A_c$ and compositions (transformed feature space). The comparison also used the solution temperature, solution time, aging temperature, aging time, and composition (original feature space). The ML models were trained and evaluated by cross-validation, and the root mean square error (RMSE) with mean and standard deviation is shown in Fig. 8c. In general, models using the transformed feature space have smaller root mean square errors than the original feature space. These ML models were then used to predict γ′ size for the 13 newly reported superalloys from published articles in the year 2023 as well as 3 superalloys that we synthesized, which were not in the dataset extracted by our pipeline (Fig. 8e–g, and Supplementary Table 4). The average of RMSE between the reported/experiments and predicted γ′ size amongst all models with transformed features is 83.00, much lower than 143.63 using original features (Fig. 8d). Such a significant increase in model performance suggests $(A_t{*}S_c{}^{0.5})^{0.5} + A_c$ to be a significant synthesis factor for γ′ size for superalloys.

We have here ignored the occurrence of actions implicitly expressed, such as "the aged samples were …". Additionally, the dependency parser of the Stanford CoreNLP package cannot accurately construct dependent relations between sample, action, and parameters under certain expressed conditions. In addition, action-tuple information distributed across two or more separate sentences are not handled. We have also not incorporated the synthesis and processing parameters that are described in the figures.

In recent years, large-scale language pretraining models, such as GPT (Generative Pretraining Transformer), have revolutionized the field of NLP[49–51]. These models are trained on vast amounts of unannotated texts and can then be fine-tuned for specific NLP tasks. Essentially, these models are creating a "well-read" black box that interprets language at a high level and can perform a multitude of tasks within that language. ChatGPT, a specific implementation of the GPT models, was trained using Reinforcement Learning from Human Feedback (RLHF) and exhibits impressive abilities in conversational interaction[52]. It can handle dialog format, answer follow-up questions, admit mistakes, and even reject inappropriate requests. However, despite these advances, ChatGPT and similar models have limitations. The sheer scale of these models necessitates substantial computational resources and vast, well-organized corpora for training, which could limit their accessibility. Additionally, these models are sensitive to input phrasing, and a slight rephrasing can yield different outputs. In the context of materials science, it is also difficult for GPT to solve the correlation between composition, synthesis, and properties, summarize the extracted database and automatically build models to mine the physical feature factors related to the target properties. The lack of complete and structured data is an issue. AI models like ChatGPT primarily learn from vast amounts of text data but do not inherently possess structured data extraction capabilities. Although they can provide general information and summarize existing knowledge, extracting specific details and organizing them into a structured database for a quantitative prediction model is a more complex task that ChatGPT cannot yet achieve for materials. Here we introduce a semi-supervised text mining method, in a small-corpus and with low costs, to extract action sequences and their parameters related to synthesis and processing conditions. This automatically forms a machine learnable dataset containing synthesis actions and parameters, chemical compositions and γ′ phase size. The dataset has then been used to capture an explicitly expressed synthesis factor for predicting γ′ phase coarsening. The synthesis factor derived from text mining significantly improves the performance of the data-driven γ′ size prediction model. This strategy is applied easily on a specific problem in order to distill synthesis actions and parameters from scratch instead of fine-tuning, or pre-training large amounts of corpora.

In conclusion, we have shown how knowledge of materials synthesis and processing in the literature can be extracted by text mining. The code for this semi-supervised text mining pipeline is available at https://github.com/MGEdata/Action_extractor. A web-based toolkit is also available at http://superalloydigger.mgedata.cn/#/spre_extractor for online use. This open-source code and toolkit can also be generalized to other alloys. As the scientific literature grows, it is inevitable that NLP will become a promising tool to extract and learn from published and unpublished work and provide a format that is machine-readable and AI-useable.

METHODS

Article retrieval and preprocessing

The scientific articles for superalloys used in this work were published before the year 2022, and full texts were automatically obtained in extensible markup language (XML) format using Elsevier's Scopus, Science Direct APIs (https://dev.elsevier.com/) and the extended scrape package of ChemDataExtractor[21]. Corpus preprocessing and table parsing was executed by SuperalloyDigger as our previous work represented[25]. Totally we achieved 16604 article corpora with more than 0.4 million paragraphs and 6644 composition tables.

Paragraph classification

To determine which paragraph contained alloy synthesis information, we manually applied binary labels to 1885 paragraphs from 80 different journal articles. The positive samples represent that the paragraphs contain synthesis and processing information, while negative samples stand for the paragraphs not related. The labeled paragraphs were split by 9:1 with 90% of the corpus for training and validation, and the remaining 10% for testing. Then a binary logistic regression classifier was trained by scikit-learn package[53], as shown in Supplementary Fig. 5a.

Three different feature extraction methods were compared during paragraph classification, namely Bag of Words, TF-IDF (term frequency–inverse document frequency), and BERT (Bidirectional Encoder Representation from Transformers). The BERT model has been pre-trained on 16604 superalloy corpora and the pre-training details are given in Supplementary Method 2. Each paragraph in the article was represented by a feature vector of Bag of Words, TF-IDF, and BERT, concatenated with a simple binary heuristic vector to distinguish whether the section title comprised keywords like "Experiments" or "Methods".

The accuracy and F1 scores by different feature extraction methods are shown in Supplementary Fig. 5b. The highest overall F1 score of 96.35% was obtained using TF-IDF.

Supplementary Fig. 5c shows the learning curves of the logistic regression model. The F1 scores can reach ~95% when the training data set size is 1800. The logistic regression model was trained on 1885 labeled paragraphs using TF-IDF as feature construction and used to predict all paragraphs in the whole corpus. After removing paragraphs with less than 20 words and those predicted as negative, it finally yielded approximately 14,487 positive paragraphs related to synthesis and processing.

Word embedding model

Two word embedding models were pre-trained on approximately 16,000 unlabeled full-text superalloy corpus by Word2Vec continuous bag of words (CBOW) and FastText[36,54]. The Word2vec model for superalloy has already been built and validated in our previous work[25]. Word2Vec can capture the semantic similarity between a word pair according to the context. If two words are semantically close, then their Word2Vec similarity is high, such as "solutioned", "aged", and "forged". For FastText, each word is represented by a sum of its character n-grams[35,36]. FastText embeddings can capture sub-word structure, multiple word senses, and uncertainty information. For example, the words with same root such as "arc-melted", "induction-melted", and "pre-melted" will have a high cosine-similarity with each other[54]. Word2vec helps find the most similar syntactic and semantic words, and FastText constructs word embeddings from character-level n-gram representations to find words with greater morphological similarity[33–36]. As shown in the Supplementary Fig. 6, a bag of n-grams as additional features was used in FastText to capture partial information about the local word order and the word with same prefix or suffix could be regarded as having similar meaning. Thus, combing the Word2vec and FastText models can help to find entities with similar syntax, semantics and morphology for synthesis and processing actions.

BiLSTM-CRF model

The BiLSTM-CRF model was also used for NER tasks. LSTM is a variant of recurrent neural network (RNN) and better at capturing both forward and backward context. The traditional softmax layer is replaced by CRF in this model to capture the interdependency of each label. To train such a BiLSTM-CRF model, we designed five entity labels: action (ACT), superalloy name (MAT), sample descriptor (DSC), material property (PRO), and property value (PV). Examples for each kind of label are given in Supplementary Method 1, along with a detailed explanation for annotation rules.

1308 sentences from 84 articles were randomly sampled from synthesis paragraphs for annotation by hand to ensure that a diverse range of synthesis and processing types were covered. All annotations were performed by a single materials scientist. During annotation, "BIO" sequence labeling method was applied.

All annotated sentences were split into training (80%), validation (10%), and testing sets (10%). The validation set was used for hyperparameter optimization evaluation and the final model achieved a total precision of 89.34%, recall of 78.30%, and F1 score of 83.46% on testing set with approximately 131 sentences (approximately 4500 words). The categorical precision, recall and F1 score for each category are shown in supplementary Table 2. For action entity, the precision, recall and F1 score are only 77.02%, 72.98% and 74.95%.

Evaluation metrics

We used precision, recall, and F1 score as the metrics to evaluate paragraph classification, NER, and dependency parsing, which are shown in below equations[55–57].

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{5}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{6}$$

$$\text{F1score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{7}$$

Interdependency resolution

To merge the extracted composition and action data fragment, we tailored some rules based on the writing habit of experts in superalloy. Firstly, the composition table was extracted to find all alloy names and compositions in the article by our previous method[25]. And then, different strategies were performed according to the condition of composition table and synthesis paragraph as follows.

If no alloy name entity is recognized in the sentence containing the action entity, but in the paragraph where this sentence is located, and the sentence preceding the sentence recognizes the alloy name entity, then the alloy name entity will be associated with the action entity.

If no alloy name is recognized in the sentence preceding the action, and any alloy name entity in the composition table does not appear in the full paragraph, we will assume that every alloy in the composition table is associated with the action entity.

Symbolic regression

To capture the explicitly expressed synthesis factor for γ′ size, we performed SR analysis by a Python library, namely gplearn. There were several hyper-parameters to optimize, including pc, ps, ph, pp, and parsimony coefficient. We applied grid search to evaluate the performance on a space containing 5 pc values, 10 ps values, 10 ph values, 2 pp values, and 3 parsimony coefficients. The detailed hyper-parameters setup can be found in the Supplementary Table 5. There were totally 3000 hyper-parameters during SR, and for each hyper-parameter, the populations evolved for 20 generations. So totally 60,000 candidate formulas were generated.

Prediction model for γ′ size

The whole dataset was randomly divided into a 70% fraction for model training and validation, and the remaining 30% fraction for model testing. Support vector regression (SVR) with linear kernel (SVR.lin) and radial basis function kernel (SVR.rbf), gradient boosting regression (GBR), Bayesian linear regression (BR), k-nearest neighbor regression (KNR), adaptive boosting regression (AdaBR), kernel ridge regression (KRR), random forest regression (RFR), stochastic gradient descend regression (SGDR), elastic net regression (ENR), and lasso regression (LR) were employed. For parameter optimization, 100 times of 5-fold cross-validation on 70% training dataset was performed. All the models were re-trained with their optimized parameters to predict the γ′ sizes for new alloys.

Superalloy synthesis and characterization

The alloys were synthesized from raw metals with a purity higher than 99.95%. 40 g ingot was prepared by vacuum arc melting by melting for eight times. The ingots were then sealed in a quartz tube with an argon atmosphere. Then, solution heat treatment at 1225 °C for 12 h was applied to all experimental alloys followed by air cooling to eliminate the composition segregation. The solutioned samples were further cut and aged at 1100 °C for 168 h followed by water cooling. All samples were obtained by the

standard metallographic method and chemically etched for seconds in a solution of HCl: H_2O: HNO_3 = 1:1:1. A Zeiss GeminiSEM 300 field-emission scanning electron microscope (SEM) in backscattered electron imaging mode was used to observe the γ/γ′ microstructure. An energy-dispersive X-ray spectroscopy detector was used to determine the alloy composition. The γ′ size of alloys are measured by a computer vision framework by Liu et al[31].

DATA AVAILABILITY

All the generated data set can be found available at https://github.com/MGEdata/Action_extractor. The DOIs of 16604 articles and 30 validated articles together with the extracted dataset containing composition, heat treatment, and γ′ size are available from https://github.com/MGEdata/Action_extractor/tree/main/Database.

CODE AVAILABILITY

All the source code used in this work is available at https://github.com/MGEdata/Action_extractor. Furthermore, a web-based toolkit has been developed; further examples of how to use and adapt the toolkit can be found at http://superalloydigger.mgedata.cn/#/spre_extractor.

Received: 2 May 2023; Accepted: 24 September 2023;
Published online: 06 October 2023

REFERENCES

1. Isayev, O. Text mining facilitates materials discovery. *Nature* **571**, 42–43 (2019).
2. Choudhary, K. et al. Recent advances and applications of deep learning methods in materials science. *npj Comput. Mater.* **8**, 59 (2022).
3. Agrawal, A. & Choudhary, A. Perspective: Materials informatics and big data: Realization of the 'fourth paradigm' of science in materials science. *APL Mater.* **4**, 053208 (2016).
4. Curtarolo, S. et al. The high-throughput highway to computational materials design. *Nat. Mater.* **12**, 191–201 (2013).
5. Ramprasad, R., Batra, R., Pilania, G., Mannodi-Kanakkithodi, A. & Kim, C. Machine learning in materials informatics: Recent applications and prospects. *npj Comput. Mater.* **3**, 54 (2017).
6. 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 Comput. Mater.* **5**, 21 (2019).
7. Zakutayev, A. et al. An open experimental database for exploring inorganic materials. *Sci. Data* **5**, 1–12 (2018).
8. Kirklin, S. et al. The Open Quantum Materials Database (OQMD): Assessing the accuracy of DFT formation energies. *npj Comput. Mater.* **1**, 15010 (2015).
9. Batra, R., Song, L. & Ramprasad, R. Emerging materials intelligence ecosystems propelled by machine learning. *Nat. Rev. Mater.* **6**, 655–678 (2021).
10. Moosavi, S. M. et al. Capturing chemical intuition in synthesis of metal-organic frameworks. *Nat. Commun.* **10**, 1–7 (2019).
11. Schweidtmann, A. M. et al. Machine learning meets continuous flow chemistry: Automated optimization towards the Pareto front of multiple objectives. *Chem. Eng. J.* **352**, 277–282 (2018).
12. MacLeod, B. P. et al. Self-driving laboratory for accelerated discovery of thin-film materials. *Sci. Adv.* **6**, eaaz8867 (2020).
13. Dragone, V., Sans, V., Henson, A. B., Granda, J. M. & Cronin, L. An autonomous organic reaction search engine for chemical reactivity. *Nat. Commun.* **8**, 1–8 (2017).
14. Li, Z. et al. Robot-Accelerated Perovskite Investigation and Discovery. *Chem. Mater.* **32**, 5650–5663 (2020).
15. Baldan, R. et al. Solutioning and aging of MAR-M247 nickel-based superalloy. *J. Mater. Eng. Perform.* **22**, 2574–2579 (2013).
16. Ramsperger, M. et al. Solution heat treatment of the single crystal nickel-base superalloy CMSX-4 fabricated by selective electron beam melting. *Adv. Eng. Mater.* **17**, 1486–1493 (2015).
17. Raccuglia, P. et al. Machine-learning-assisted materials discovery using failed experiments. *Nature* **533**, 73–76 (2016).
18. Kim, E. et al. Machine-learned and codified synthesis parameters of oxide materials. *Sci. Data* **4**, 170127 (2017).
19. Court, C. J. & Cole, J. M. Auto-generated materials database of Curie and Néel temperatures via semi-supervised relationship extraction. *Sci. data* **5**, 1–12 (2018).
20. Kumar, P., Kabra, S. & Cole, J. M. Auto-generating databases of Yield Strength and Grain Size using ChemDataExtractor. *Sci. Data* **9**, 1–11 (2022).
21. Swain, M. C. & Cole, J. M. ChemDataExtractor: A toolkit for automated extraction of chemical information from the scientific literature. *J. Chem. Inf. Model.* **56**, 1894–1904 (2016).
22. Krallinger, M., Rabal, O., Lourenço, A., Oyarzabal, J. & Valencia, A. Information retrieval and text mining technologies for chemistry. *Chem. Rev.* **117**, 7673–7761 (2017).
23. Kim, E. et al. Inorganic materials synthesis planning with literature-trained neural networks. *J. Chem. Inf. Model.* **60**, 1194–1201 (2020).
24. Kim, E., Huang, K., Jegelka, S. & Olivetti, E. Virtual screening of inorganic materials synthesis parameters with deep learning. *npj Comput. Mater.* **3**, 53 (2017).
25. Wang, W. et al. Automated pipeline for superalloy data by text mining. *npj Comput. Mater.* **8**, 1–12 (2022).
26. Weston, L. et al. Named entity recognition and normalization applied to large-scale information extraction from the materials science literature. *J. Chem. Inf. Model.* **59**, 3692–3702 (2019).
27. Huo, H. et al. Semi-supervised machine-learning classification of materials synthesis procedures. *npj Comput. Mater.* **5**, 1–7 (2019).
28. Ruan, J. et al. Accelerated design of novel W-free high-strength Co-base superalloys with extremely wide γ/γ′ region by machine learning and CALPHAD methods. *Acta Mater.* **186**, 425–433 (2020).
29. Liu, Y. et al. Predicting creep rupture life of Ni-based single crystal superalloys using divide-and-conquer approach based machine learning. *Acta Mater.* **195**, 454–467 (2020).
30. Liu, P. et al. Machine learning assisted design of γ′-strengthened Co-base superalloys with multi-performance optimization. *npj Comput. Mater.* **6**, 1–9 (2020).
31. Liu, P. et al. Evolution analysis of γ′ precipitate coarsening in Co-based superalloys using kinetic theory and machine learning. *Acta Mater.* **235**, 118101 (2022).
32. O'Gorman, T. et al. MS-MENTIONS: Consistently Annotating Entity Mentions in Materials Science Procedural Text. *EMNLP 2021 - 2021 Conf. Empir. Methods Nat. Lang. Process. Proc.* 1337–1352. https://doi.org/10.18653/v1/2021.emnlp-main.101 (2021).
33. Mikolov, T., Chen, K., Corrado, G. & Dean, J. Efficient estimation of word representations in vector space. *1st Int. Conf. Learn. Represent. ICLR 2013 - Work. Track Proc.* 1–12 (2013).
34. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. & Dean, J. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* 3111–3119 (2013).
35. Athiwaratkun, B., Wilson, A. G. & Anandkumar, A. Probabilistic fasttext for multi-sense word embeddings. ACL 2018 - 56th Annu. Meet. Assoc. Comput. Linguist. Proc. Conf. (Long Pap. **1**, 1–11 (2018).
36. Joulin, A. et al. FastText.zip: Compressing text classification models. 1–13 (2016).
37. Riloff, E. & Jones, R. Learning dictionaries for information extraction by multi-level bootstrapping. *Proc. Natl. Conf. Artif. Intell.* 474–479 (1999).
38. Riloff, E. & Phillips, W. An Introduction to the Sundance and AutosSlog Systems. *Tech. Rep. UUCS-04-015, Sch. Comput. Univ. Utah* 1–47 (2004).
39. De Marneffe, M. C., MacCartney, B. & Manning, C. D. Generating typed dependency parses from phrase structure parses. *Proc. 5th Int. Conf. Lang. Resour. Eval. Lr. 2006* 449–454 (2006).
40. Manning, C. et al. The Stanford CoreNLP Natural Language Processing Toolkit. 55–60 https://doi.org/10.3115/v1/p14-5010 (2015).
41. Chen, Y. et al. Development of low-density γ/γ′ Co–Al–Ta-based superalloys with high solvus temperature. *Acta Mater.* **188**, 652–664 (2020).
42. Lass, E. A., Sauza, D. J., Dunand, D. C. & Seidman, D. N. Multicomponent γ′-strengthened Co-based superalloys with increased solvus temperatures and reduced mass densities. *Acta Mater.* **147**, 284–295 (2018).
43. Makineni, S. K., Nithin, B. & Chattopadhyay, K. Synthesis of a new tungsten-free γ-γ′ Cobalt-based superalloy by tuning alloying additions. *Acta Mater.* **85**, 85–94 (2015).
44. Stephens, T. gplearn. https://gplearn.readthedocs.io/en/latest/intro.html.
45. Ardell, A. J. The effect of volume fraction on particle coarsening: theoretical considerations. *Acta Metall.* **20**, 61–71 (1972).
46. Lifshitz, I. M. & Slyozov, V. V. The kinetics of precipitation from supersaturated solid solutions. *J. Phys. Chem. Solids* **19**, 35–50 (1961).
47. Wagner, C. Theory of the aging of precipitation by dissolution (Ostwald maturation). *Rep. Bunsen Soc. Phys. Chem.* **65**, 581–591 (1961). http://onlinelibrary.wiley.com/doi/10.1002/bbpc.19610650704/abstract.
48. Calderon, H. A., Voorhees, P. W., Murray, J. L. & Kostorz, G. Ostwald ripening in concentrated alloys. *Acta Metall. Mater.* **42**, 991–1000 (1994).
49. OpenAI. *GPT-4 Tech. Report.* **4**, 1–100 (2023).
50. Koubaa, A. GPT-4 vs. *GPT-3.5: A Concise Showdown*. 1–5. https://doi.org/10.20944/preprints202303.0422.v1 (2023)

51. Brown, T. B. et al. Language models are few-shot learners – special version. *Adv. Neural Inf. Process. Syst.* 2020 (2020).
52. Shen, Y. et al. ChatGPT and Other Large Language Models Are Double-edged Swords. *Radiology* **307**, (2023).
53. Baranwal, A., Bagwe, B. R. & M, V. *Mach. Learn. Python.* **12**, 128–154 (2019).
54. Thavareesan, S. & Mahesan, S. Sentiment Lexicon Expansion using Word2vec and fastText for Sentiment Prediction in Tamil texts. *MERCon 2020 - 6th Int. Multidiscip. Moratuwa Eng. Res. Conf. Proc.* 272–276 https://doi.org/10.1109/MERCon50084.2020.9185369 (2020).
55. Goutte, C. & Gaussier, E. A probabilistic interpretation of precision, recall and F-score, with implication for evaluation. In *European conference on information retrieval* 345–359 (Springer, 2005).
56. Sniegula, A., Poniszewska-Mararida, A. & Chomatek, L. Study of named entity recognition methods in biomedical field. *Procedia Comput. Sci.* **160**, 260–265 (2019).
57. Japkowicz, N. Why question machine learning evaluation methods. In *AAAI workshop on evaluation methods for machine learning* 6–11 (2006).

ACKNOWLEDGEMENTS

This work is financially supported by the National Key Research and Development Program of China (2021YFB3702403, 2022YFB3707502), National Natural Science Foundation of China (52201061, U22A20106), Fundamental Research Funds for the Central Universities (FRF-TP-22-008A1), USTB MatCom of Beijing Advanced Innovation Center for Materials Genome Engineering, and the CNNC Science Fund for Talented Young Scholars (FY222506000902).

AUTHOR CONTRIBUTIONS

J.X. conceived the project. The study was designed by J.X., Y.S. and T.L., text mining programs were performed by W.W., X.J. and S.T., the manuscript prepared by W.W., X.J., S.T., Y.S., T.L. and J.X. All authors discussed the results and commented on the manuscript.

COMPETING INTERESTS

The authors declare no competing interests.

ADDITIONAL INFORMATION

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41524-023-01138-w.

Correspondence and requests for materials should be addressed to Xue Jiang, Turab Lookman or Yanjing Su.