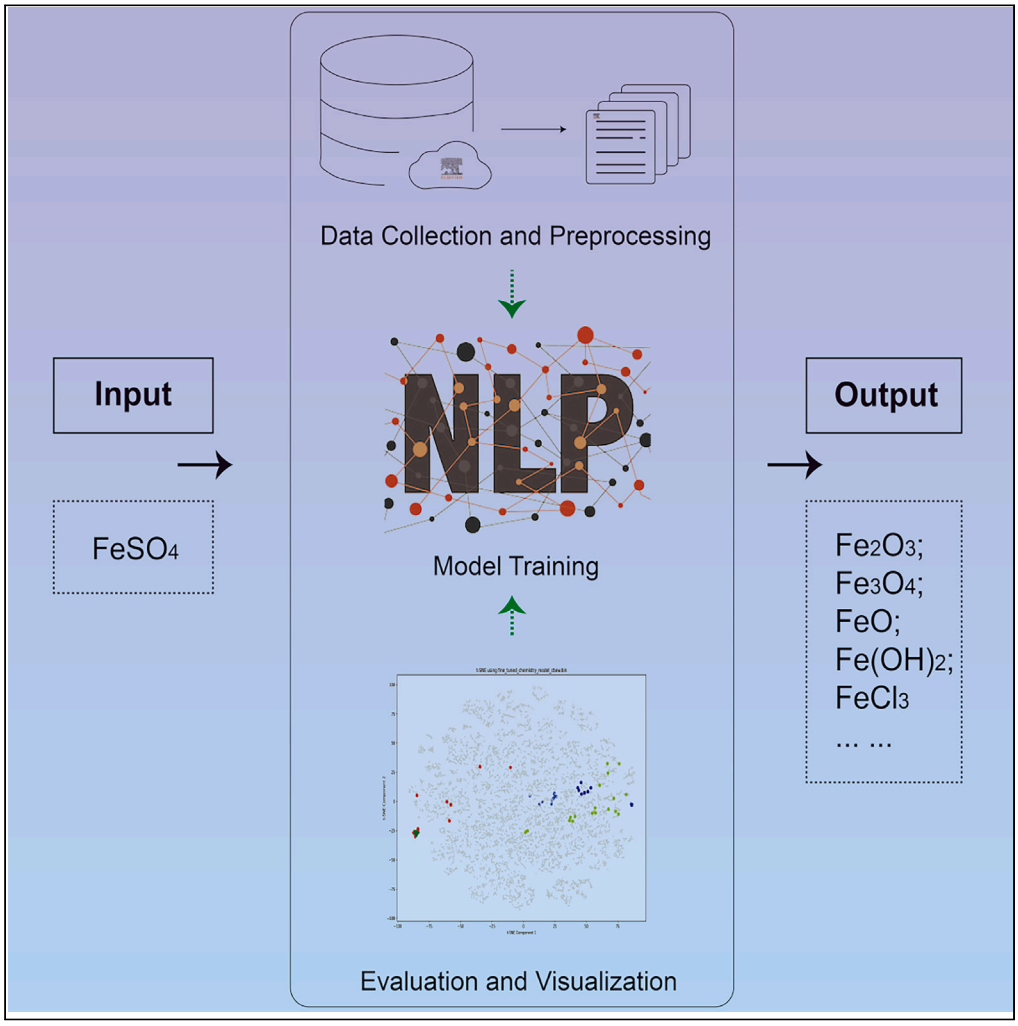
iScience

CellPress
OPEN ACCESS

Article

Enhancing chemical synthesis research with NLP: Word embeddings for chemical reagent identification—A case study on nano-FeCu



Dingding Cao, Mieow Kee Chan

mkchan@segi.edu.my

Highlights

NLP models identify chemical reagents for materials synthesis

Specialized corpus boosts chemical term recognition

ChemFastText-Tuned outperforms in chemical synonym analysis

Cao & Chan, iScience 27, 110780
October 18, 2024
https://doi.org/10.1016/j.isci.2024.110780

iScience

CellPress
OPEN ACCESS

Article

Enhancing chemical synthesis research with NLP: Word embeddings for chemical reagent identification—A case study on nano-FeCu

Dingding Cao[1,2] and Mieow Kee Chan[1,3,*]

SUMMARY

Nanoparticle synthesis is complex, influenced by multiple variables including reagent selection. This study introduces a specialized corpus focused on "Fe, Cu, synthesis" to train a domain-specific word embedding model using natural language processing (NLP) in an unsupervised environment. Evaluation metrics included average cosine similarity, visual analysis via t-distributed stochastic neighbor embedding (t-SNE), synonym analysis, and analogy reasoning analysis. Results indicate a strong correlation between learning rate and cosine similarity, with enhanced chemical specificity in the tailored model compared to general models. The framework facilitates rapid identification of potential reagents for nano-FeCu synthesis, enhancing precision in nanomaterial research. This innovative approach offers a data-driven pathway for chemical material synthesis, demonstrating significant interdisciplinary applications.

INTRODUCTION

Over recent years, nanotechnology research has grown exponentially due to its unique properties such as super-paramagnetism and excellent catalytic reactivity in addressing the complex challenges across various disciplines. As an emerging interdisciplinary field, it encompasses physics,[1] chemistry,[2] and biology[3] for its applications in medicine, energy, and materials science.[4] In the medical field, nanomedicine offers a reliable way to improve the efficiency and precision of drug delivery and significantly improve therapeutic outcomes.[5] In the energy field, nanomaterials demonstrated large electrochemical active surfaces and distinctive optical and electronic characteristics. These are crucial for the development of next-generation energy storage applications.[6]

In nanomaterials, nano-FeCu is a particularly interesting bimetallic nanomaterial that exhibits a unique combination of high catalytic performance and reactive activity. The excellent catalytic performance of nano-FeCu is primarily attributed to its increased surface area which allows efficient electron transfer.[7] Nano-FeCu bimetallic nanoparticles exhibit high potential in wastewater treatment, pollutant degradation, and heavy metal ion removal.[8] Previous studies done by Chan et al. demonstrated that nano-FeCu removed ammonia via the oxidation-reduction process[9,10] and organic matter was removed by the combination effect of adsorption and reductive reactions.[11,12]

Nano-FeCu can be synthesized using two primary approaches, which are top-down and bottom-up methods.[13] The top-down approach starts with bulk materials, reducing them to the nanoscale using techniques such as lithography or etching, allowing for precise control over the nanoparticles' size and shape. On the other hand, the bottom-up approach synthesis of nano-FeCu particles from atomic or molecular precursors through chemical reactions, such as the simultaneous reduction of iron (III) and copper (II) salts in the presence of a reducing agent like sodium borohydride.[14] This results in well-defined chemical compositions and the potential for self-assembly into complex structures.[13]

However, the main challenge is to obtain the desired size and shape of nano-FeCu, which requires precise control of the synthesis conditions. The synthesis method, temperature, duration, and selection of chemicals are crucial in the chemical reduction method.[9] These factors influence the morphology, size, and properties of nanoparticles. Temperature affects the kinetics of particle growth, where minor fluctuations in temperature result in significant disparities in particle size. Meanwhile, the choice of chemical reagents directly affects the chemical structure and functionality of nanoparticles.[15] Additionally, Retana et al. (2020) found that the presence of complexing agents such as trisodium citrate and EDTA affected the properties of iron nanoparticles.[16] Given the variety of chemical materials and experimental conditions, it is important for material scientists to develop a reliable technique for nanoparticle synthesis.

Owing to the diversity of synthesis methods, researchers often encounter difficulties in identifying ideal chemical reagents and synthesis conditions. A tedious trial-and-error exploratory approach is required to determine the right synthesis methods for a novel bimetallic nanoparticle pairing for a new application.[17] Despite the numerous studies reported by the researchers on nanoparticle synthesis methods, the current methodology continues to depend on heuristic and empirical approaches and lacks effective design guidelines.[18] It is desired to

[1]Centre for Water Research, Faculty of Engineering, Built Environment and Information Technology, SEGi University. Jalan Teknologi, Kota Damansara, Petaling Jaya 47810, Selangor Darul Ehsan, Malaysia
[2]Department of Electrical and Electronic Engineering, Guangdong Technology College, Zhaoqing 526100, China
[3]Lead contact
*Correspondence: mkchan@segi.edu.my
https://doi.org/10.1016/j.isci.2024.110780

develop a rapid and cost-effective guide to assist the researchers in identifying the most appropriate chemical reagents for a specific application.

In recent years, the development in artificial intelligence (AI), especially in natural language processing (NLP) and large-scale model architecture has shown remarkable enhancement due to the advances in computing power and algorithms.[19] The unveiling of ChatGPT by OpenAI marks the commencement of a new era in AI-driven large language models (LLMs).[20] Text mining methods within the general domain of NLP have made substantial progress in recent years, where it is able to understand and analyze a wide variety of textual data with high efficiency.[21] There has been a significant expansion of NLP technologies in chemistry and materials science research.[22] For instance, NLP-enabled extraction of chemical reactions and material properties from vast scientific literature significantly accelerates data aggregation and research processes in these fields.[23] Named entity recognition (NER) is a fundamental method in NLP that can be used to extract information from material science literature, assisting researchers to obtain knowledge from the pool of scientific publications accurately and rapidly.[24] For instance, applying NER techniques can enable the automated extraction of data concerning nanomaterials, such as their synthesis methods, properties, and applications.[25]

Creating domain-specific datasets is a critical step in applying AI-related technologies to the field of chemistry. However, creating a nanomaterial synthesis conditions query system that organizes data such as the choice of chemical reagents, synthesis duration, temperature variations, and synthesis methods, from published papers is a magnificent task. This is because the chemical reagents for nano-synthesis, the methods, and the properties of nanomaterials are normally presented in unstructured form in the scientific literature. Manual extraction of such data is labor-intensive and impractical.[26] Efforts were made by adopting NER to extract information from research papers; however, its accuracy is questionable. This is due to the complex notation in chemical formulas, ambiguity in authors' descriptions or methodologies of reaction processes, lack of extraction of entity relationships, and errors in automated or manual annotation. The cumulative errors from multiple conditions eventually lead to poor practicality of the system, limiting the advancement of NLP technology in the field of material synthesis.[27]

Word embedding is a technique in NLP that transforms words into a continuous vector space, capturing the inherent semantic relationships. This approach leverages the text information from both the syntactic and semantics context of the training data. By analyzing word co-occurrences in a text corpus, it computes the vectors for each word based on the co-occurrence of words in a text corpus.[28] It can effectively learn the semantic relationships between words from large or specialized unstructured textual data, thereby capturing the implicit semantics of those words. Typically, embedding models are trained in an unsupervised or self-supervised manner, meaning that there is no need to label the dataset in advance. It also addressed the issue of limited available annotated data for training.[29] Word embeddings led to significant improvements in many downstream tasks, such as speech recognition,[30] machine translation,[31] and text spelling disambiguation.[32] They also have wide applications in part-of-speech tagging and sentiment analysis.[33]

Word embedding models can be trained using different algorithms. Currently, commonly used word vector models include Word2Vec, as proposed by Mikolov et al.,[34] global vectors for word representation (GloVe) introduced by Stanford University,[35] bidirectional encoder representations from transformers (BERT),[36] and generative pre-trained transformer (GPT)[37] promoted by Google AI and OpenAI. Each of these models adopts different network structures and optimization strategies. Word2Vec, trained by shallow neural networks, is adept at capturing semantic and syntactic relationships, but its drawback is that it considers only word-level information, overlooks polysemy, and neglects phrase or sentence-level information. GloVe trains by factorizing the word co-occurrence matrix and can effectively capture linear relationships between words. However, it requires the storage and processing of large word co-occurrence matrices. BERT, characterized by its deep, bidirectional transformer structure, considers the contextual information of words but requires extensive computational resources and has a long pre-training time.

FastText is a framework proposed by the Facebook research team for text classification and word embedding.[38] The FastText model focuses primarily on subword information, allowing the model to better account for morphological variations within languages. By providing embeddings for each subword of a word, FastText can more flexibly handle rare or novel vocabulary. Given its subword-level representation, FastText can generate vectors for any word, including the new words.[39] The architecture of FastText is concise and efficient, particularly suitable for large textual data and exceptionally effective in performing text classification.[40] In addition, its versatile embedding mechanism allows FastText models to capture rich semantic information and contextual nuances within texts.

In this study, FastText model was chosen for training as it provides a comprehensive and accurate representation of chemical vocabulary. A corpus related to the three key terms "Fe, Cu, synthesis" was proposed for unsupervised training of word vectors, and a chemical word embedding model for a specific scientific field (Fe, Cu, synthesis) was developed by NLP technology. The finding was compared with OpenAI's text-embedding-ada-002 word embedding model, based on the transformer architecture and with the BERT model. This research combines traditional experimental research which involved hands-on laboratory work and empirical analysis with modern machine learning techniques, to provide information on potential chemical reagents for the synthesis of nano-FeCu particles.

RESULTS AND DISCUSSION

Dataset analysis

Figure 1 shows the Schematic of the experiment workflow. Figure 2 presents an analysis of the sentence length and word frequency in the Chem_Dataset. The occurrence of each element from the periodic table in the dataset is depicted in Figure 2A. The figures clearly showed that the occurrence of iron and copper significantly exceeded the other elements. Following closely in the frequency are carbon, oxygen, and hydrogen, consistent with their prevalence in actual chemical reactions.[41] The statistical findings validate the specialization and credibility of the developed dataset, Chem_Dataset, particularly in the domain of iron and copper.

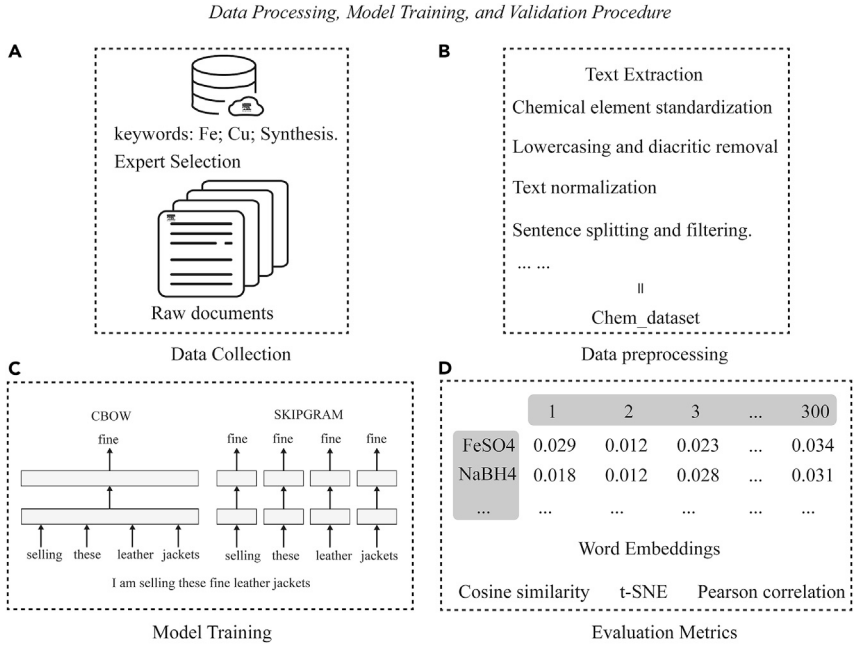


Figure 1. Schematic of experiment workflow

(A–D) Workflow of (A) data collection, (B) data preprocessing, (C) model training, and (D) evaluation metrics.

Figure 2B illustrates the length of each sentence in the Chem_Dataset Approximately 95% of sentences were less than 100 words, and it displayed a normal distribution. This finding is in agreement with Gennaro et al.,[42] where appropriate sentence length provides rich contextual information, aiding the capture of semantic relations between words, and thereby aiding in good understanding and processing of text

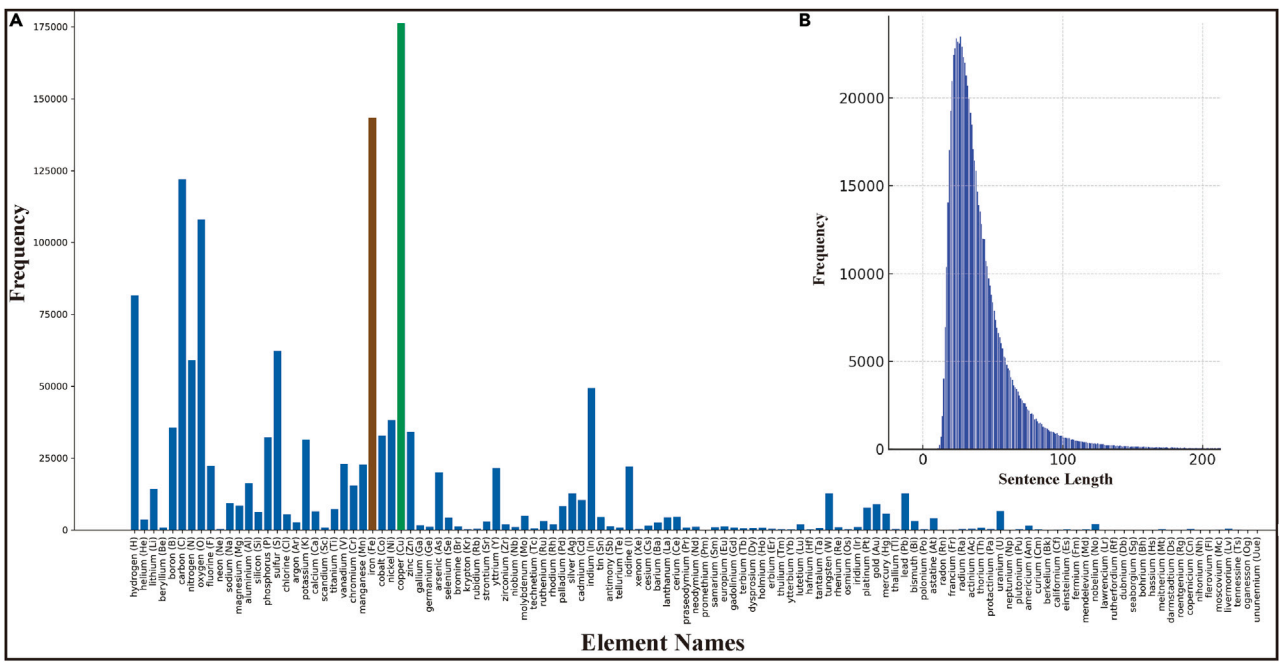


Figure 2. Analysis of sentence length and elemental occurrence within the Chem_Dataset

(A) Elemental occurrence distribution.

(B) Sentence length distribution.

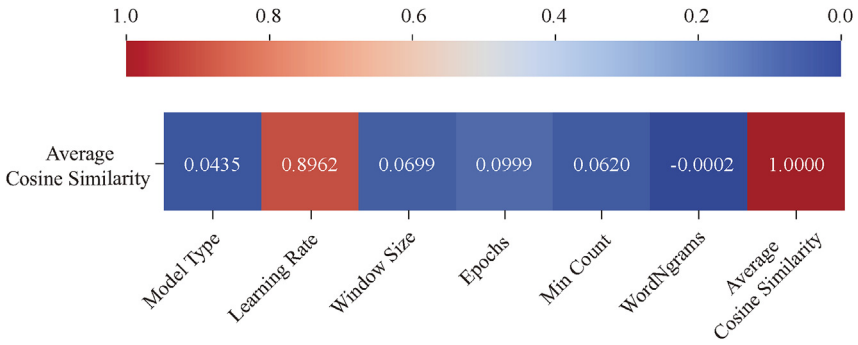


Figure 3. Correlational analysis of grid search parameters and average cosine similarity

The figure illustrates the Pearson correlation coefficient analysis between various hyperparameters and average cosine similarity.

data in NLP applications. The data collected in this study could be used for embedding model training and other upstream NLP tasks such as chemical word classification, relationship extraction, and NER.

Analysis of average cosine similarity

Figure 3 depicts a Pearson correlation coefficient analysis between the hyperparameters involved in the grid search and the average cosine similarity. It is observed that the learning rate exhibits a strong correlation with average cosine similarity ($r = 0.8962$). This could be attributed to the learning rate's pivotal role in controlling the convergence speed of the model during training.[43] The impact of the remaining parameters on average cosine similarity is negligible ($r < 0.1$). The details of the data can be found in Data S12 (Average Cosine Similarity Results.xlsx).

Based on the analytical outcomes of grid search model results, higher scores within a determined range on the positive dataset (D1 and D2) signify superior model performance, owing to the considerable correlational significance of these character pairs. For instance, one pair in D1 is ['ethanol', 'methanol'], and another in D2 is ['lignin', 'cellulose']. However, when the model achieves excessively high scores, such as those greater than 0.99, it may be indicative of overfitting (refer to Data S12 Model ID: 1, 10, 15, 18…). Based on the dataset and common knowledge, "ethanol" is inherently "ethanol," and despite demonstrating similar characteristics to "methanol," it is implausible for their correlation to exceed 0.99.[44]

Conversely, lower scores within a certain range on the negative dataset (D3 and D4) are preferable as these character pairs might not possess any chemical correlation. For instance, a pair in D3 is ['sodium chloride', 'petroleum'], and in D4 is ['ethanol', 'eagle']. The manifestation of overfitting characteristics by the model will also reflect abnormally high scores, greater than 0.95, on the negative dataset (refer to Data S12 Model ID: 1, 10, 15, 18…). Based on the dataset, a correlation greater than 0.95 between "ethanol" and "eagle" indicates that the model has overfitted and no longer holds practical application value as a chemical word vector model.[45]

Table 1 shows the average cosine similarity test results of different models on the constructed positive datasets D1 and D2, and negative datasets D3 and D4. The FastText-Base model exhibits the lowest average cosine similarity scores on positive datasets D1 and D2, at 0.6940 and 0.5717, respectively. The ChemFastText-Tuned model achieves the highest scores on positive datasets D1 and D2, reaching 0.9556 and 0.8583 respectively, surpassing the GPT-ada-002 model's performance of 0.8996 on dataset D1. From the perspective of the dataset, the FastText-Base model serves as the baseline model and is trained only on the Wikipedia 2017, UMBC web base corpus, and statmt.org news dataset, without training on the Chem_Dataset, as a result, it reflects the lowest scores on the test sets.

However, despite not being specifically trained on the Chem_Dataset, the GPT-ada-002 model still manages to exhibit commendable scores on D1 and D2. This can be attributed to the GPT-ada-002 word embedding model being trained on a more extensive and diversified dataset.[46] Furthermore, the ChemFastText-Tuned model, by approximatively training on the Wikipedia 2017, UMBC webbase corpus, statmt.org news dataset, and Chem_Dataset, achieves the best average cosine similarity scores. This highlights the enhancing capability of specialized datasets on the model and the necessity for the construction of professional datasets.[47]

Moreover, the performances of the ChemFastText-Opt and ChemFastText-Tuned models reveal that when scores on the positive dataset (D1 and D2) increase, the scores on the negative dataset (D3 and D4) also rise. This is indeed plausible as the amount of training data increases.[48] For instance, two terms, even if unrelated from a chemical perspective, may exhibit certain relatedness if the training corpus consists of fairy tales.

The employment of diverse training corpora in pre-training phases has been quantitatively shown to contribute to an improvement in model robustness.[49] However, it could be difficult to collect diverse data in a niche research area. This study showed that the fine-tuned model, ChemFastText-Tuned, surpassing the ChemFastText-Opt model (D1–D4) and GPT-ada-002 model (D1) in terms of average cosine similarity scores. This indicated that in highly specialized fields, well-performing word vector models can still be trained through fine-tuning when the training data are in the absence of diversity.

Table 1. Comparative evaluation of average cosine similarity scores across validation sets (D1–D4) for different models

	Average cosine similarity scores			
Model name	D1	D2	D3	D4
FastText-Base	0.6940	0.5717	0.2115	0.0907
ChemFastText-Opt	0.9240	0.7635	0.6623	0.6135
ChemFastText-Tuned on Chem_Dataset	0.9556	0.8583	0.7947	0.7536
GPT-ada-002	0.8996	0.8584	0.7986	0.7840

Average cosine similarity scores for FastText-Base, ChemFastText-Opt, ChemFastText-Tuned, and GPT-ada-002 models on positive (D1 and D2) and negative (D3 and D4) datasets.

As shown in Table 2, the performance of the ChemFastText-Tuned model across various datasets illustrates the specific impacts of the range and quality of training data on model effectiveness. Specifically, after training on a specialized chemical dataset (Chem_Dataset), the model achieved high average cosine similarity scores of 0.9556 and 0.8583 on chemically related datasets D1 and D2, respectively, showcasing its capability to identify and align relevant chemical terms.[50]

When the model's training data included a broader range of chemical knowledge and some noise (ChemEnhanced_Dataset), its scores slightly decreased to 0.9528 and 0.8506 on D1 and D2, respectively, indicating that the introduction of more contextual and background information slightly affected the model's ability to process chemical terms. With even more noise in the training dataset (NoiseSimulated_Dataset), the scores on D1 and D2 further dropped to 0.9515 and 0.8480, respectively, further confirming the negative impact of increased noise on the model's accuracy in identifying chemical terms. However, the increase in noise somewhat helped the model to distinguish completely unrelated terms, as shown by the scores dropping to 0.7770 and 0.7258 on D3 and D4, respectively, suggesting that the model's ability to differentiate chemically unrelated terms might have been enhanced. These results emphasize the importance of carefully selecting and optimizing training data in chemical NLP applications. To enhance the model's generalizability and reduce overfitting, the introduction of a certain amount of noise is beneficial, but excessive noise can weaken the model's performance within its specialty area. Effective data management and model training strategies should aim to find the optimal balance between these factors, ensuring that the model can precisely process specialized terms while being adaptable to a broader range of applications.[51]

As shown in Table 3, the performance of the BERT-Tuned model across different datasets at various learning rates was evaluated. When the BERT model was adjusted at a high learning rate of 1×10^{-3} across different datasets, it scored 1.0000 on all datasets (D1–D4), indicating the model's inability to effectively differentiate between datasets, which may be due to undertraining leading to model underfitting. High scores on chemically related datasets (D1 and D2) and unrelated datasets (D3 and D4) demonstrate that the model was unable to correctly evaluate the differences between chemically related and unrelated word pairs, thus failing to provide practicality for chemical reagent identification.

At lower learning rates (1×10^{-4} and 1×10^{-5}), the average cosine similarity scores of the BERT-Tuned model decreased sequentially from D1 to D4. Specifically, for Chem_Dataset, the scores decreased from 0.8531 in D1 to 0.7083 in D4 at a learning rate of 1×10^{-4}, reflecting the model's improved ability to discriminate term relevance based on chemical context. This pattern remained consistent as the learning rate decreased, highlighting the impact of learning rate adjustments on model sensitivity and specificity.[52]

Comparing these results with the performance of the ChemFastText-Tuned on Chem_Dataset model, which achieved scores of 0.9556 and 0.8583 in D1 and D2 respectively, it is evident that the BERT-Tuned model may require careful optimization to match the specificity achieved by FastText. The inherent differences in model architecture (BERT's deep bidirectionality gives it greater contextual awareness) suggest that while BERT might offer deeper semantic insights, it is also more susceptible to overfitting, hence the need for careful adjustment of hyperparameters to optimize performance.[53]

As analyzed in Figure 4, the memory and CPU utilization during the training of fine-tuned FastText and BERT models under the Chem_Dataset were compared. It was observed that the fine-tuning of the FastText model required less than 150 min, whereas the fine-tuning of the BERT model required nearly ten times longer under equivalent resource conditions. This indicates that under CPU conditions, the cost

Table 2. Comparison of average cosine similarity scores across different datasets for ChemFastText-Tuned models

	Average cosine similarity scores			
Model name	D1	D2	D3	D4
ChemFastText-Tuned on Chem_Dataset	0.9556	0.8583	0.7947	0.7536
ChemFastText-Tuned on ChemEnhanced_Dataset	0.9528	0.8506	0.7801	0.7384
ChemFastText-Tuned on NoiseSimulated_Dataset	0.9515	0.8480	0.7770	0.7258

Average cosine similarity scores for ChemFastText-Tuned models trained on Chem_Dataset, ChemEnhanced_Dataset, and NoiseSimulated_Dataset on datasets D1-D4.

Table 3. Comparative performance of BERT-Tuned Models across different datasets at various learning rates

		Average cosine similarity scores			
Learning Rates	Model name	D1	D2	D3	D4
1×10^{-3}	BERT-Tuned on Chem_Dataset	1.0000	1.0000	1.0000	1.0000
1×10^{-4}		0.8531	0.7935	0.7238	0.7083
1×10^{-5}		0.8545	0.7913	0.7224	0.6634
1×10^{-3}	BERT-Tuned on ChemEnhanced_Dataset	1.0000	1.0000	1.0000	1.0000
1×10^{-4}		0.8667	0.7966	0.7449	0.7037
1×10^{-5}		0.8536	0.7886	0.7161	0.6525
1×10^{-3}	BERT-Tuned on NoiseSimulated_Dataset	1.0000	1.0000	1.0000	1.0000
1×10^{-4}		0.8463	0.7768	0.7083	0.6829
1×10^{-5}		0.8537	0.7860	0.7113	0.6387

Average cosine similarity scores for BERT-Tuned models trained on Chem_Dataset, ChemEnhanced_Dataset, and NoiseSimulated_Dataset at different learning rates.

of fine-tuning FastText was significantly lower than that of the BERT model. Moreover, the utilization of hardware resources was higher for FastText, with CPU usage reaching up to 80%, compared to an average utilization of 60% for the BERT model.

Regarding memory usage, the fine-tuned FastText occupied five times the memory compared to the fine-tuned BERT, whose memory usage was notably lower and maintained a relatively stable curve throughout the training period. This indicates that despite BERT's high computational complexity, its memory consumption was optimized, possibly due to more efficient batch processing and memory management strategies implemented in its execution.

It is noteworthy that the use of graphics processing units (GPUs) could significantly shorten the training time of the BERT model, as the parallel computing capabilities of GPUs can accelerate the training process of the BERT model substantially. However, considering the impact of computational resource requirements, this study focused on the lighter model, FastText.[54]

Visual analysis of t-SNE

To effectively evaluate and compare the performances of different word embedding models, this section utilizes nano-FeCu as a case study to validate the classification capabilities of models. The selection of chemical reagents was strategically aligned with the common components used in the bottom-up synthesis of nano-FeCu.[55] This methodical choice ensures the evaluation and assessment carried out are reflective of real-world applications and are thus, highly relevant, and contextual to the synthesis process of nano-FeCu.[56]

Figures 5A–5D depicts the t-SNE visual classification results of chemicals generated by FastText-Base, GPT-ada-002, ChemFastText-Opt, and ChemFastText-Tuned word embedding models. The chemical word vector distribution created by the FastText-Base model is relatively scattered, as shown in Figure 5A. This implied that this model has difficulty distinctly categorizing different types of chemicals. Similarly, the image presented in Figure 5B showed that the GPT-ada-002 model manages to effectively cluster chemicals, but it still demonstrates limited capability in distinguishing the properties of different chemical reagents. This could be observed by the intermediate blue and yellow color distribution Figure 5B. This is because both FastText-Base and GPT-ada-002 models were not trained on the specialized chemical datasets (Chem_Dataset). This implied that a large-scale generic object model such as FastText-Base and GPT-ada-002 may

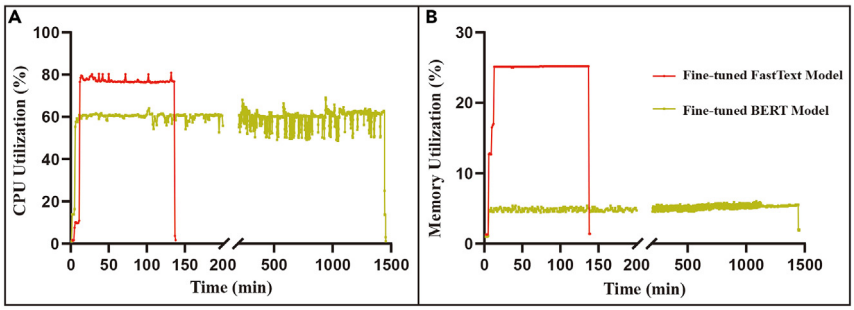


Figure 4. Comparison of CPU and memory utilization for fine-tuned FastText and BERT-Tuned models during training

(A) CPU usage over time shows higher utilization for Fine-Tuned FastText compared to BERT. (B) Memory usage over time indicates FastText occupies more memory than BERT.

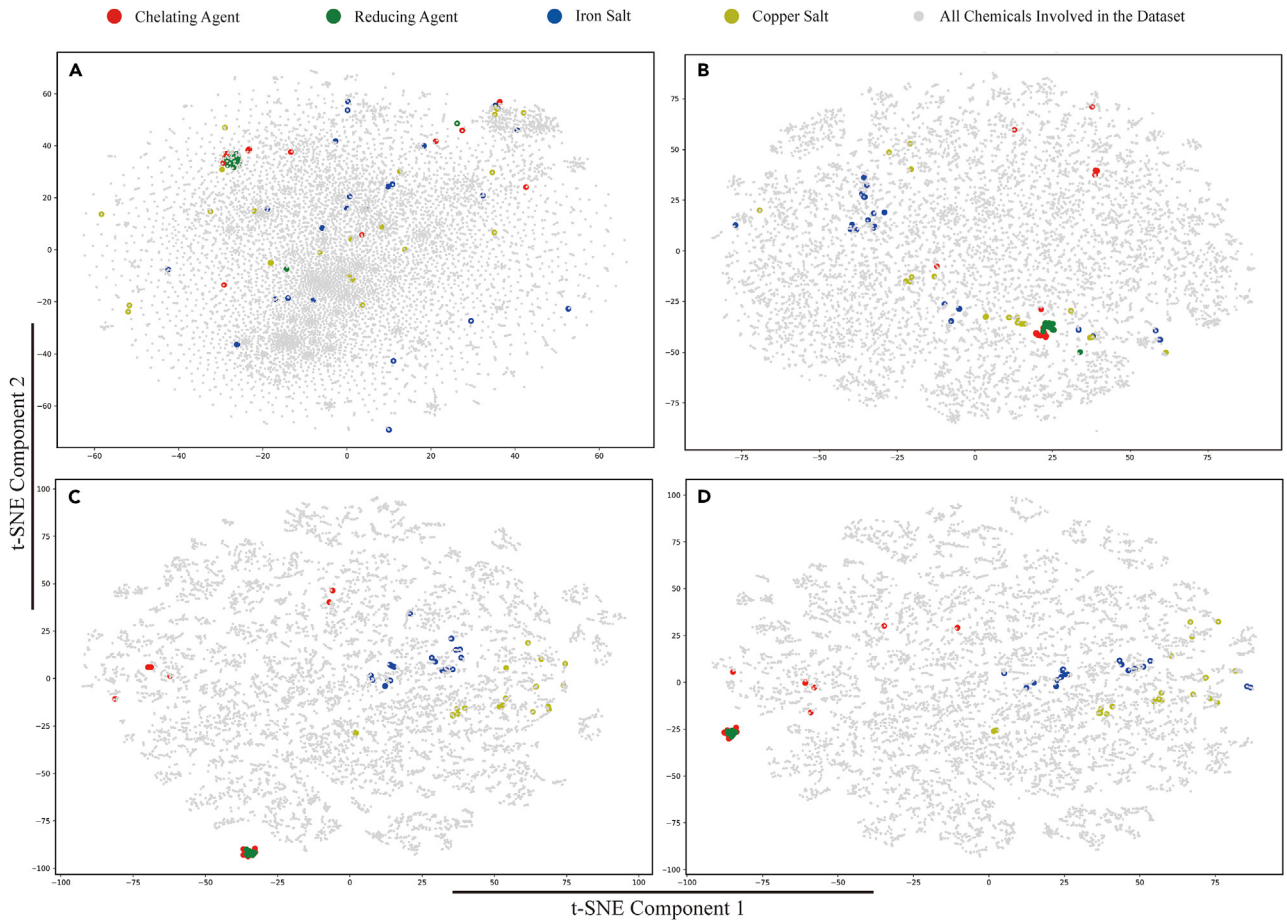


Figure 5. Analysis of word embeddings, t-SNE visualization of various types of chemical reagents using different models

(A) FastText-Base.

(B) GPT-ada-002.

(C and D) ChemFastText-Opt and (D) ChemFastText-Tuned. Visualization of t-SNE results for chemical reagents using (A) FastText-Base, (B) GPT-ada-002, (C) ChemFastText-Opt, and (D) ChemFastText-Tuned models, illustrating the clustering capabilities of each model.

have difficulty in understanding and processing of chemical-related vocabulary and concepts due to limited training on chemical-related database.[57]

In Figures 5C and 5D, the models ChemFastText-Opt and ChemFastText-Tuned exhibit excellent clustering capabilities and effectively distinguish between copper salts and iron salts. This can be seen from the color distribution in Figures 5C and 5D where the colors are more focused in a particular area. Notably, some reducing agents and chelating agents are clustered together, which was indicated by the overlapping of red and green color areas in Figures 5C and 5D. This occurrence may be attributed to the relatively low frequency of these terms within the original dataset, resulting in insufficient learning and optimization during the training phase. Consequently, the positioning of these chemical terms within the word vector space is imprecise. This phenomenon implies the importance of incorporating extensive and diverse training data when constructing word embedding models to ensure a more refined and accurate representation of terms.[58]

Synonym analysis

In this section, keywords such as hydrothermal method and $FeSO_4$, presented in Table 4 are selected, and potential pertinent information is explored through nearest neighbor searches with k = 250. The key findings are tabulated in Table 4.

The FastText-Base model has difficulty extracting useful information/words for the keyword "hydrothermal method." This implies that the basic FastText model was inadequate for capturing the relations between intricate chemical terminologies within the chemical domain (Chem_Dataset). Meanwhile, both the ChemFastText-Opt and ChemFastText-Tuned models successfully identified various synthesis techniques related to the hydrothermal method, such as "solvothermal," "gel-hydrothermal," and "co-hydrothermal." This denotes the models' ability to recognize synthesis methods analogous to the hydrothermal method, reflecting their broad comprehension of the contextual background related to hydrothermal synthesis techniques.

Table 4. Assessment of chemical term similarities and relationships in embedding models

Selected keywords	Model name	Top-250 nearest neighbors (ranked in descending order, with meaningless characters manually removed and normalized, and the selected meaningful words are displayed below)
hydrothermal method	FastText-Base	No meaningful words were located.
	ChemFastText-Opt	gel-hydrothermal; solvo-hydrothermal; co-hydrothermal; hydro-thermal; gel method; precipitation method; mechanosynthesized; aluminothermic; thermocatalysis; hydrometallurgy
	ChemFastText-Tuned	Solvothermal; gel-hydrothermal; co-hydrothermal; hygrothermal; mechanosynthesized; aluminothermic; superthermal; precipitation method; thermolysis; gel method
	GPT-ada-002	Gel-hydrothermal; Solvo-hydrothermal; Hydrolysis; Hydrometallurgical; Electrothermal; Pyrohydrometallurgy; Hydrotreatment; Electrolysis; Solvothermal; Gasification; Photocatalysis; Hydrogenation; Distillation
$FeSO_4$	FastText-Base	$FeCl_2$; $FeCl_3$; FeCl; $FeCO_3$; Fe2; Fe3; FeOOH; Fe_2O_3; FeS; FeS_2;
	ChemFastText-Opt	FeO_4; FeK_2O_4; $FeMn_2O_4$; Fe_3S_4; FeN_2O_6; Fe_2LiO_4; FeS_4; FeO_4P;
	ChemFastText-Tuned	$FeCl_3$; $FePO_4$; $FeSO_4$; Fe_2O_3; $Fe_3O_8P_2$; Fe_2LiO_4; Fe_2S_3; FeO; $Fe(NO_3)_3$; $Fe(OH)_2$; $Fe(acac)_3$
	GPT-ada-002	$FeSO_4 \cdot 7H_2O$; Fe_2O_3; Fe_3O_4; FeO; $Fe_2O_3 \cdot SiO_2$; $Fe(OH)_2$; Fe_2O_4Si; Fe_2LiO_4; FeS
$CuCl_2$	FastText-Base	CuCl; $CuSO_4$; CuI; Cu2; CuBr; CuS; CuII; 2Cu; CuO
	ChemFastText-Opt	$CuSO_4$; $Cu(OTf)_2$; $Cu(NO_3)_2$; $CuCO_3$; $Cu(OH)_2$; $Cu(OAC)_2$; CuF_2; $Cu4O_3$; CuO; CuS_3
	ChemFastText-Tuned	$Cu(OTf)_2$; $Cu(Im)_2$; $CuNO_3$; $CuCO_3$; $Cu(OH)_2$; CuF_2; $CuSO_4$; $Cu(OAc)_2$; CuF_3; $CuMnO_2$; CuO_2
	GPT-ada-002	CuCl; $CuCl_2 \cdot H_2O$; $CuCl_2 \cdot 4H_2O$; $CuCl_2 \cdot 6H_2O$; Cu (II)Cl; Cu(ntb)Cl; CuL_2Cl; Cu (II)L1Cl; CuO
$NaBH_4$	FastText-Base	$LiAlH_4$; $SnCl_2$; borohydride; Na_2SO_3; $LiBH_4$; SmI_2; dithionite; 2-propanol; N_2H_4; H_3PO_3; HCOOH
	ChemFastText-Opt	NH_3; N_2H_4; C_2H_5; H_2; N_2O_5;
	ChemFastText-Tuned	$LiAlH_4$; NaN_3; NH_4; N_2H_4; $NaHSO_3$; $NaClO_3$;
	GPT-ada-002	AlH_4Na; NaH; sodiumborohydride; borohydride; NH_3; NaOH;
EDTA	FastText-Base	EGTA; BAPTA; 1,10-phenanthroline; 8-hydroxyquinoline; ethylenediamine; Chelex; Desferal; calcein; ionomycin
	ChemFastText-Opt	EDDHA; EDDA; TMS-EDTA; K_2EDTA; Ethylenediamine; EGTA; TEPA; TETA; Diaminetetraacetic acid; ED3A; DTPA; DOTA;
	ChemFastText-Tuned	EDDHA; EDDA; TMS-EDTA; K_2EDTA; Ethylenediamine; EGTA; TEPA; TETA; EDAA; EDPA; DTPA; DO3A-AMBA; H4EDTA; Gd-EOB-DTPA
	GPT-ada-002	EDTA-Na; Na_2EDTA; EDTA-2Na; H4EDTA; EGTA; ethylenediamine; DTPA; DTPA-Zn, DTPA-Fe; diethylenetriamine; CDTA; HEDP

Top-250 nearest neighbors for selected keywords identified by FastText-Base, ChemFastText-Opt, ChemFastText-Tuned, and GPT-ada-002 models.

Some terms like "aluminothermic," "superthermal," and "thermocatalysis" are associated with thermal reactions or high-temperature conditions, suggesting the possibility of their utilization under conditions similar to the "hydrothermal method." Notably, the term "solvothermal" provided by ChemFastText-Tuned displays an environment similar to the hydrothermal method, with a variance in the solvent used, highlighting the model's acute perception of subtle differences. This indicated the importance of training models with the relevant dataset which was tailored to the chemical field in this study.

The GPT-ada-002 model also identified synthesis and treatment methods related to the hydrothermal method in its neighboring word options, such as "hydrolysis" and "hydrometallurgical," revealing its ability to recognize processes involving water related to the hydrothermal method. This model also recognized several methods related to chemical treatment, synthesis, and separation, like "electrothermal," "gasification," "photocatalysis," and "hydrogenation." These techniques are not synonymous with the hydrothermal method, but they are some of the key methods for chemical treatment and synthesis. This indicates that the performance of the ChemFastText-Tuned model is comparable with the GPT-ada-002 model on the keyword "hydrothermal method." Proper training data are vital to enhance the performance of the word embedding model.[59]

Table 5. Model performance on D5 dataset (Top-50 accuracy %)

Model Name	Top-50 accuracy (%)
ChemFastText-Tuned on Chem_Dataset	7.08
ChemFastText-Tuned on ChemEnhanced_Dataset	7.96
ChemFastText-Tuned on NoiseSimulated_Dataset	5.31
BERT-Tuned	Not applicable

Top-50 accuracy percentages for analogy reasoning analysis using ChemFastText-Tuned models trained on Chem_Dataset, ChemEnhanced_Dataset, and NoiseSimulated_Dataset, along with BERT-Tuned model results.

For the ChemFastText-Opt, ChemFastText-Tuned, and GPT-ada-002, all evaluated words produced results of substantial relevance across the three models. Moreover, not only were synonyms discovered, but also reagents exhibiting analogous chemical properties were identified. For instance, multiple iron salts were found in the term "$FeSO_4$," and various copper salts were detected in "$CuCl_2$." These findings suggest that the method of seeking synonyms serves as a viable approach for identifying chemically analogous reagents.[60,61] In the identification of $FeSO_4$, FastText-Base primarily displays various chemical forms related to iron, such as $FeCl_3$ and Fe_2O_3, highlighting different valence states of iron and its compounds with other elements. Additionally, the ChemFastText-Tuned seems to offer a more extensive and pertinent range of synonyms, likely attributed to the model undergoing more meticulous chemical-specific tuning. Taking '$CuCl_2$" as an example, both ChemFastText-Opt and ChemFastText-Tuned accurately present Cu-related compounds.

However, ChemFastText-Tuned offers more specific compounds, such as Cu $(Im)_2$ and $CuMnO_2$, showing its enhanced specialization. GPT-ada-002, on the other hand, goes a step further, offering specific hydrated forms like $FeSO_4 \cdot 7H_2O$, demonstrating its profound understanding of the chemical structure of substances. In the case of "EDTA," all four models provided other chelating agents similar in chemical structure or function. The ChemFastText-Tuned model excels in this regard, listing not only common EDTA-related compounds but also identifying more complex chemical derivatives like DO3A-AMBA and Gd-EOB-DTPA, demonstrating higher sensitivity and depth of knowledge. This reveals that model fine-tuning can amplify its recognition capabilities for complex systems.

Similarly, the GPT-ada-002 model also exhibits commendable performance in certain aspects. In the identification of $FeSO_4$, FastText-Base primarily displays various chemical forms related to iron, such as $FeCl_3$ and Fe_2O_3, highlighting different valence states of iron and its compounds with other elements. GPT-ada-002, on the other hand, goes a step further, offering specific hydrated forms like $FeSO_4 \cdot 7H_2O$, demonstrating its profound understanding of the chemical structure of substances.

$NaBH_4$ is a widely used reducing agent. From the results, it can be observed that FastText-Base provides a series of reducing agents similar in chemical properties to $NaBH_4$, such as $LiAlH_4$ and dithionite. ChemFastText-Tuned goes further, recognizing compounds with similar electronegativity or those that might play a role in reactions, such as NaN_3 and $NaHSO_3$. GPT-ada-002 introduces the core structure of borohydride, proving its commendable chemical discernment. In conclusion, the chemical word embedding models can capture the latent relationships and properties between chemical reagents, displaying them as the closest neighbors in high-dimensional space. This synonym analysis not only displays the capabilities of different models but also provides chemists with a tool for swiftly locating substances or methods related to a given chemical substance or method. This holds immense potential value in literature searches, new material designs, or predictions of chemical reactions.

Analogy reasoning analysis

In the analysis presented in Table 5, the performance of the ChemFastText-Tuned model on the D5 Dataset for Analogy Reasoning was displayed across different datasets. It was observed that the overall top-50 accuracy remained below 10%, which may be attributed to the specialization of the model when fine-tuned using a dataset specific to a particular domain.[62,63] This specialization, while enhancing the model's performance for certain tasks or data types, also resulted in the sacrifice of its general capabilities. Notably, the accuracy improved from 7.08 to 7.96 when additional chemically relevant knowledge was incorporated. This outcome underscores the importance of mixing training data. By retaining specialized datasets and integrating broader general data, the model can maintain its expertise while also learning and understanding a wider range of contexts and applications. Furthermore, when the model was fine-tuned using the NoiseSimulated Dataset, the accuracy decreased to 5.31. This decrease was mainly because the D5 test set predominantly pertained to chemical and general knowledge, with less involvement of musical knowledge. It is important to note that BERT is primarily designed to handle contextual word vectors rather than static word meanings. For analogy tasks, models like GloVe or Word2Vec might perform better as they provide more consistent spatial properties to support analogical reasoning. In this study, the use of BERT-Tuned did not yield meaningful content for Analogy Reasoning. Research by Ascari et al.,[64] supports these observations, noting that while BERT excels in tasks requiring deep semantic understanding, it often struggles with analogical reasoning due to the variability introduced by context-specific embeddings.

Conclusion

This research combines NLP technology and specialized knowledge in the field of chemistry, establishing a specialized corpus focused on "Fe, Cu, synthesis" and training domain-specific word embedding models in an unsupervised environment. This provides a novel, data-driven strategy and perspective for research in the field of chemical synthesis. Several words embedding models, including FastText-Base, ChemFastText-Opt, ChemFastText-Tuned, BERT-Tuned, and GPT-ada-002, were thoroughly evaluated to explore the relationships and similarities between chemical terminologies. The results illustrate that models trained and tuned with specialized chemical knowledge, especially the ChemFastText-Tuned model, exhibit exceptional performance in exploring chemical terminologies and identifying related chemical processes. This model can not only identify compounds with similar chemical attributes and functions but also present a broader range of specific and relevant compounds, demonstrating profound chemical knowledge and refined chemical insights. The methodology used in this research provides a cross-disciplinary approach, enabling chemists to quickly locate chemical reagents or methods related to a given chemical substance or method, with immense potential value for literature searches, new material design, and more. Although this study primarily focuses on identifying chemical reagents related to FeCu synthesis, the methodological framework and technical workflow can be readily adapted to other material synthesis domains.

Limitations and future work

This study focuses on training chemical word embedding models to recommend reagents with similar chemical properties. FeCu bimetallic nanoparticle is selected as the case study and limited to the keywords of the material and synthesis method. Similar approach can be done to other material or chemical process.

One limitation of this study stems from the construction of the corpus, where the quality of the training data derived from PDF files depends on the accuracy of data extraction during the data processing phase.

However, predefined filtering criteria could be added to process the collected articles to enhance data quality and model reliability.

FastText is selected as the baseline rather than BERT for comparison purpose in this study due to its computational efficiency and acceptable accuracy. More other architectures could be considered in the future work, in addition to FastText to minimize potential bias due to structure architecture.

The accuracy of the models was evaluated in terms of cosine similarity and synonym analysis, which are specifically designed to test the efficacy of the word embeddings in a chemical context. In future work, more downstream tasks such as text classification, NER, and questions and answer approach are recommended for comprehensive evaluation of word embedding models.

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources should be directed to the lead contact, Mieow Kee Chan (mkchan@segi.edu.my).

Materials availability

This study did not generate new unique reagents.

Data and code availability

- All supplemental materials and data related to this study have been deposited at Zenodo and are publicly available as of the date of publication. The DOI is listed in the key resources table.
- The complete codebase used for text preprocessing, model training, and validation in this study has been deposited at Zenodo and is publicly available as of the date of publication. The DOI is listed in the key resources table.
- Any additional information required to reanalyze the data reported in this paper is available from the lead contact upon request.

ACKNOWLEDGMENTS

We express our heartfelt appreciation for the invaluable contributions made by the late ChanChin Wang. His lasting impact on our research community will always be remembered. The research funding provided by SEGi University via grant number SEGiIRF/2022-Q2/FoEBEIT/011 is greatly appreciated.

AUTHOR CONTRIBUTIONS

M.K.C.: writing – review & editing, visualization, validation, supervision, resources, project administration, investigation, funding acquisition, and conceptualization. D.C.: visualization, methodology, investigation, formal analysis, writing – original draft, and writing – review & editing.

DECLARATION OF INTERESTS

The authors declare no competing interests.

DECLARATION OF GENERATIVE AI AND AI-ASSISTED TECHNOLOGIES IN THE WRITING PROCESS

During the preparation of this work the author(s) used ChatGPT and Grammarly in order to enhance the clarity and readability of the manuscript. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

STAR★METHODS

Detailed methods are provided in the online version of this paper and include the following:

- KEY RESOURCES TABLE
- METHOD DETAILS
- DATA COLLECTION
- DATA PREPROCESSING
- CONSTRUCTION OF THE VALIDATION SETS
- TRAINING AND VALIDATION OF WORD EMBEDDINGS MODELS
  - t-Distributed stochastic neighbor embedding (t-SNE)
- SYNONYM ANALYSIS
- ANALOGY REASONING ANALYSIS

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at https://doi.org/10.1016/j.isci.2024.110780.

Received: January 12, 2024
Revised: March 11, 2024
Accepted: August 16, 2024
Published: August 29, 2024

REFERENCES

1. Li, A., Wei, H., Cotrufo, M., Chen, W., Mann, S., Ni, X., Xu, B., Chen, J., Wang, J., Fan, S., et al. (2023). Exceptional points and non-Hermitian photonics at the nanoscale. Nat. Nanotechnol. *18*, 706–720. https://doi.org/10.1038/s41565-023-01408-0.
2. Du, J., Zeng, L., Yan, T., Wang, C., Wang, M., Luo, L., Wu, W., Peng, Z., Li, H., and Zeng, J. (2023). Efficient solvent- and hydrogen-free upcycling of high-density polyethylene into separable cyclic hydrocarbons. Nat. Nanotechnol. *18*, 772–779. https://doi.org/10.1038/s41565-023-01429-9.
3. Mavridi-Printezi, A., Guernelli, M., Menichetti, A., and Montalti, M. (2020). Bio-Applications of Multifunctional Melanin Nanoparticles: From Nanomedicine to Nanocosmetics. Nanomaterials *10*, 2276. https://doi.org/10.3390/nano10112276.
4. Malik, S., Muhammad, K., and Waheed, Y. (2023). Nanotechnology: A Revolution in Modern Industry. Molecules *28*, 661. https://doi.org/10.3390/molecules28020661.
5. Sahu, T., Ratre, Y.K., Chauhan, S., Bhaskar, L.V.K.S., Nair, M.P., and Verma, H.K. (2021). Nanotechnology based drug delivery system: Current strategies and emerging therapeutic potential for medical science. J. Drug Deliv. Sci. Technol. *63*, 102487. https://doi.org/10.1016/j.jddst.2021.102487.
6. Das, H., Pathak, B., Khanam, S., Kalita, P.K., and Datta, P. (2022). Nanomaterials for next generation energy storage applications. MRS Commun. *12*, 285–294. https://doi.org/10.1557/s43579-022-00193-6.
7. Nguyen, T.B., Dong, C.-D., Huang, C.P., Chen, C.-W., Hsieh, S.-L., and Hsieh, S. (2020). Fe-Cu bimetallic catalyst for the degradation of hazardous organic chemicals exemplified by methylene blue in Fenton-like reaction. J. Environ. Chem. Eng. *8*, 104139. https://doi.org/10.1016/j.jece.2020.104139.
8. Xia, Q., Zhang, D., Yao, Z., and Jiang, Z. (2022). Revealing the enhancing mechanisms of Fe–Cu bimetallic catalysts for the Fenton-like degradation of phenol. Chemosphere *289*, 133195. https://doi.org/10.1016/j.chemosphere.2021.133195.
9. Chan, M.K., Abdullah, N., Rageh, E.H.A., Kumaran, P., and Tee, Y.S. (2021). Oxidation of ammonia using immobilised FeCu for water treatment. Sep. Purif. Technol. *254*, 117612. https://doi.org/10.1016/j.seppur.2020.117612.
10. Kee, C.M., Mun, N.K., Kumaran, P., Selvam, R., Kumaran, R., Raja, S.D., and Shen, T.Y. (2021). The impact of ammonia concentration and reducing agents on the ammonia oxidation performance of embedded nano-FeCu. Mater. Chem. Phys. *274*, 125189. https://doi.org/10.1016/j.matchemphys.2021.125189.
11. Chan, M.K., Kumaran, P., Thomas, X.V., Natasha, E., Tee, Y.S., Mohd Aris, A., Ho, Y.P., and Khor, B.C. (2023). Embedded nanoFeCu for sewage treatment: Laboratory-scale and pilot studies. Can. J. Chem. Eng. *101*, 3751–3758. https://doi.org/10.1002/cjce.24721.
12. Chan, M.K., Lim, K.W., Mohd Aris, A., Ho, Y.P., and Khor, B.C. (2023). Flow rate impact on the performance of immobilized nanoFeCu for sewage treatment and its reusability. Can. J. Chem. Eng. *101*, 6404–6416. https://doi.org/10.1002/cjce.24926.
13. Baig, N., Kammakakam, I., and Falath, W. (2021). Nanomaterials: a review of synthesis methods, properties, recent progress, and challenges. Mater. Adv. *2*, 1821–1871. https://doi.org/10.1039/D0MA00807A.
14. Loza, K., Heggen, M., and Epple, M. (2020). Synthesis, Structure, Properties, and Applications of Bimetallic Nanoparticles of Noble Metals. Adv. Funct. Mater. *30*, 1909260. https://doi.org/10.1002/adfm.201909260.
15. Mitchell, M.J., Billingsley, M.M., Haley, R.M., Wechsler, M.E., Peppas, N.A., and Langer, R. (2021). Engineering precision nanoparticles for drug delivery. Nat. Rev. Drug Discov. *20*, 101–124. https://doi.org/10.1038/s41573-020-0090-8.
16. RETANA, F., KHARISSOV, B., PEÑA, Y., GÓMEZ, I., and SERRANO, T. (2020). EFFECT OF COMPLEXING AGENTS ON PROPERTIES AND STABILITY OF FeS2 NANOPARTICLES. Chalcogenide Lett. *17*, 353–360. https://doi.org/10.15251/CL.2020.177.353.
17. Ramanathan, S., Gopinath, S.C.B., Arshad, M.K.M., Poopalan, P., and Perumal, V. (2021). Nanoparticle synthetic methods: strength and limitations. In Nanoparticles in Analytical and Medical Devices (Elsevier), pp. 31–43. https://doi.org/10.1016/B978-0-12-821163-2.00002-9.
18. Liu, H., Zhang, H., Wang, J., and Wei, J. (2020). Effect of temperature on the size of biosynthesized silver nanoparticle: Deep insight into microscopic kinetics analysis. Arab. J. Chem. *13*, 1011–1019. https://doi.org/10.1016/j.arabjc.2017.09.004.
19. Min, B., Ross, H., Sulem, E., Veyseh, A.P.B., Nguyen, T.H., Sainz, O., Agirre, E., Heintz, I., and Roth, D. (2024). Recent Advances in Natural Language Processing via Large Pre-trained Language Models: A Survey. ACM Comput. Surv. *56*, 1–40. https://doi.org/10.1145/3605943.
20. Liu, Y., Han, T., Ma, S., Zhang, J., Yang, Y., Tian, J., He, H., Li, A., He, M., Liu, Z., et al. (2023). Summary of ChatGPT-Related research and perspective towards the future of large language models. Meta-Radiology *1*, 100017. https://doi.org/10.1016/j.metrad.2023.100017.
21. Khurana, D., Koli, A., Khatter, K., and Singh, S. (2023). Natural language processing: state of the art, current trends and challenges. Multimed. Tool. Appl. *82*, 3713–3744. https://doi.org/10.1007/s11042-022-13428-4.
22. Lee, J.H., Lee, M., and Min, K. (2023). Natural Language Processing Techniques for Advancing Materials Discovery: A Short Review. Int. J. of Precis. Eng. and Manuf. -Green. Tech. *10*, 1337–1349. https://doi.org/10.1007/s40684-023-00523-6.
23. Butler, K.T., Davies, D.W., Cartwright, H., Isayev, O., and Walsh, A. (2018). Machine learning for molecular and materials science. Nature *559*, 547–555. https://doi.org/10.1038/s41586-018-0337-2.
24. Corbett, P., and Boyle, J. (2018). Chemlistem: chemical named entity recognition using recurrent neural networks. J. Cheminf. *10*, 59. https://doi.org/10.1186/s13321-018-0313-8.
25. Weston, L., Tshitoyan, V., Dagdelen, J., Kononova, O., Trewartha, A., Persson, K.A., Ceder, G., and Jain, A. (2019). Named Entity Recognition and Normalization Applied to Large-Scale Information Extraction from the

Materials Science Literature. J. Chem. Inf. Model. *59*, 3692–3702. https://doi.org/10.1021/acs.jcim.9b00470.
26. Hiszpanski, A.M., Gallagher, B., Chellappan, K., Li, P., Liu, S., Kim, H., Han, J., Kailkhura, B., Buttler, D.J., and Han, T.Y.-J. (2020). Nanomaterial Synthesis Insights from Machine Learning of Scientific Articles by Extracting, Structuring, and Visualizing Knowledge. J. Chem. Inf. Model. *60*, 2876–2887. https://doi.org/10.1021/acs.jcim.0c00199.
27. Wang, L., Gao, Y., Chen, X., Cui, W., Zhou, Y., Luo, X., Xu, S., Du, Y., and Wang, B. (2023). A corpus of CO2 electrocatalytic reduction process extracted from the scientific literature. Sci. Data *10*, 175. https://doi.org/10.1038/s41597-023-02089-z.
28. Yin, D., Wu, Z., Yokota, K., Matsumoto, K., and Shibayama, S. (2023). Identify novel elements of knowledge with word embedding. PLoS One *18*, e0284567. https://doi.org/10.1371/journal.pone.0284567.
29. Meng, Y., Huang, J., Wang, G., Wang, Z., Zhang, C., and Han, J. (2020). Unsupervised Word Embedding Learning by Incorporating Local and Global Contexts. Front. Big Data *3*, 9. https://doi.org/10.3389/fdata.2020.00009.
30. Chuang, S.-P., Liu, A.H., Sung, T.-W., and Lee, H.y. (2021). Improving Automatic Speech Recognition and Speech Translation via Word Embedding Prediction. IEEE/ACM Trans. Audio Speech Lang. Process. *29*, 93–105. https://doi.org/10.1109/TASLP.2020.3037543.
31. Chen, Q. (2023). A Smaller and Better Word Embedding for Neural Machine Translation. IEEE Access *11*, 40770–40778. https://doi.org/10.1109/ACCESS.2023.3270171.
32. Jia, L., Tang, J., Li, M., You, J., Ding, J., and Chen, Y. (2021). TWE-WSD: An effective topical word embedding based word sense disambiguation. CAAI Trans. Intell. Technol. *6*, 72–79. https://doi.org/10.1049/cit2.12006.
33. Chiche, A., and Yitagesu, B. (2022). Part of speech tagging: a systematic review of deep learning and machine learning approaches. J. Big Data *9*, 10. https://doi.org/10.1186/s40537-022-00561-y.
34. Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space.
35. Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) (Association for Computational Linguistics), pp. 1532–1543. https://doi.org/10.3115/v1/D14-1162.
36. Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of naacL-HLT, *1*, p. 2.
37. Alec, R., Karthik, N., Tim, S., and Ilya, S. (2018). Improving Language Understanding by Generative Pre-Training.
38. Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2016). Enriching Word Vectors with Subword Information. Transactions of the association for computational linguistics *5*, 135–146.
39. Khasanah, I.N. (2021). Sentiment Classification Using fastText Embedding and Deep Learning Model. Procedia Comput. Sci. *189*, 343–350. https://doi.org/10.1016/j.procs.2021.05.103.
40. Joulin, A., Grave, E., Bojanowski, P., and Mikolov, T. (2016). Bag of Tricks for Efficient Text Classification. Preprint at arXiv. https://doi.org/10.48550/arXiv.1607.01759.
41. YETTER, R.A., DRYER, F.L., and RABITZ, H. (1991). A Comprehensive Reaction Mechanism For Carbon Monoxide/Hydrogen/Oxygen Kinetics. Combust. Sci. Technol. *79*, 97–128. https://doi.org/10.1080/00102209108951759.
42. Di Gennaro, G., Buonanno, A., and Palmieri, F.A.N. (2021). Considerations about learning Word2Vec. J. Supercomput. *77*, 12320–12335. https://doi.org/10.1007/s11227-021-03743-2.
43. Na, G.S. (2022). Efficient learning rate adaptation based on hierarchical optimization approach. Neural Network. *150*, 326–335. https://doi.org/10.1016/j.neunet.2022.02.014.
44. Ibrahim, M., and Koch, B. (2015). Assessment and Mapping of Groundwater Vulnerability Using SAR Concentrations and GIS: A Case Study in Al-Mafraq, Jordan. J. Water Resour. Protect. *07*, 588–596. https://doi.org/10.4236/jwarp.2015.77047.
45. Chacko, R., Jain, D., Patwardhan, M., Puri, A., Karande, S., and Rai, B. (2020). Data based predictive models for odor perception. Sci. Rep. *10*, 17136. https://doi.org/10.1038/s41598-020-73978-1.
46. OpenAI. (2023). Introducing text and code embeddings. Open https://openai.com/index/introducing-text-and-code-embeddings/.
47. Asudani, D.S., Nagwani, N.K., and Singh, P. (2023). Impact of word embedding models on text analytics in deep learning environment: a review. Artif. Intell. Rev. *56*, 1–81. https://doi.org/10.1007/s10462-023-10419-1.
48. Lederer, A., Capone, A., Umlauft, J., and Hirche, S. (2020). How Training Data Impacts Performance in Learning-based Control. IEEE Control Systems Letters *5*, 905–910.
49. Vulić, I., Mrkšić, N., Reichart, R., Ó Séaghdha, D., Young, S., and Korhonen, A. (2017). Morph-fitting: Fine-Tuning Word Vector Spaces with Simple Language-Specific Rules. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (Association for Computational Linguistics), pp. 56–68. https://doi.org/10.18653/v1/P17-1006.
50. Schröder, S., Schulz, A., Kenneweg, P., Feldhans, R., Hinder, F., and Hammer, B. (2021). Evaluating Metrics for Bias in Word Embeddings. Preprint at arXiv. https://doi.org/10.48550/arXiv.2111.07864.
51. Montesinos López, O.A., Montesinos López, A., and Crossa, J. (2022). Overfitting, Model Tuning, and Evaluation of Prediction Performance. In Multivariate Statistical Machine Learning Methods for Genomic Prediction (Springer International Publishing), pp. 109–139. https://doi.org/10.1007/978-3-030-89010-0_4.
52. Zhang, T., Kishore, V., Wu, F., Weinberger, K.Q., and Artzi, Y. (2019). BERTScore: Evaluating Text Generation with BERT. Preprint at arXiv. https://doi.org/10.48550/arXiv.1904.09675.
53. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., and Polosukhin, I. (2017). Attention Is All You Need (Advances in Neural Information Processing Systems).
54. Choi, H., and Lee, J. (2021). Efficient Use of GPU Memory for Large-Scale Deep Learning Model Training. Appl. Sci. *11*, 10377. https://doi.org/10.3390/app112110377.
55. Wang, Y., Qiao, L., Zhang, X., Liu, Z., Li, T., and Wang, H. (2024). Green synthesis of FeCu@biochar nanocomposites through a mechanochemical method for enhanced tetracycline degradation via peroxymonosulfate activation. Sep. Purif. Technol. *328*, 125077. https://doi.org/10.1016/j.seppur.2023.125077.
56. Kolluru, A., Shuaibi, M., Palizhati, A., Shoghi, N., Das, A., Wood, B., Zitnick, C.L., Kitchin, J.R., and Ulissi, Z.W. (2022). Open Challenges in Developing Generalizable Large-Scale Machine-Learning Models for Catalyst Discovery. ACS Catal. *12*, 8572–8581. https://doi.org/10.1021/acscatal.2c02291.
57. White, A.D. (2023). The future of chemistry is language. Nat. Rev. Chem *7*, 457–458. https://doi.org/10.1038/s41570-023-00502-0.
58. Tinn, R., Cheng, H., Gu, Y., Usuyama, N., Liu, X., Naumann, T., Gao, J., and Poon, H. (2023). Fine-tuning large neural language models for biomedical natural language processing. Patterns *4*, 100729. https://doi.org/10.1016/j.patter.2023.100729.
59. Murakami, R., and Chakraborty, B. (2022). Investigating the Efficient Use of Word Embedding with Neural-Topic Models for Interpretable Topics from Short Texts. Sensors *22*, 852. https://doi.org/10.3390/s22030852.
60. Chen, Z., He, Z., Liu, X., and Bian, J. (2018). Evaluating semantic relations in neural word embeddings with biomedical and general domain knowledge bases. BMC Med. Inf. Decis. Making *18*, 65. https://doi.org/10.1186/s12911-018-0630-x.
61. Ibrahim, M., Gauch, S., Salman, O., and Alqahtani, M. (2021). An automated method to enrich consumer health vocabularies using GloVe word embeddings and an auxiliary lexical resource. PeerJ. Comput. Sci. *7*, e668. https://doi.org/10.7717/peerj-cs.668.
62. Johnson, S.J., Murty, M.R., and Navakanth, I. (2023). A detailed review on word embedding techniques with emphasis on word2vec. Multimed. Tool. Appl. *83*, 37979–38007. https://doi.org/10.1007/s11042-023-17007-z.
63. Brandl, S., Lassner, D., Baillot, A., and Nakajima, S. (2022). Domain-Specific Word Embeddings with Structure Prediction. Transactions of the Association for Computational Linguistics *11*, 320–335.
64. Ascari, R., Giabelli, A., Malandri, L., Mercorio, F., and Mezzanzanica, M. (2024). A Fistful of Vectors: A Tool for Intrinsic Evaluation of Word Embeddings. Cognit. Comput. *16*, 949–963. https://doi.org/10.1007/s12559-023-10235-3.
65. Swain, M.C., and Cole, J.M. (2016). ChemDataExtractor: A Toolkit for Automated Extraction of Chemical Information from the Scientific Literature. J. Chem. Inf. Model. *56*, 1894–1904. https://doi.org/10.1021/acs.jcim.6b00207.
66. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Müller, A., Nothman, J., Louppe, G., et al. (2012). Scikit-learn: Machine Learning in Python. J. Mach. Learn. Res. *12*, 2825–2830.
67. Elsevier (2023). Elsevier Developer Portal (Elsevier).
68. Zhou, K., Ethayarajh, K., Card, D., and Jurafsky, D. (2022). Problems with Cosine as a Measure of Embedding Similarity for High Frequency Words. Preprint at arXiv. https://doi.org/10.48550/arXiv.2205.05092.
69. Shirkhorshidi, A.S., Aghabozorgi, S., and Wah, T.Y. (2015). A Comparison Study on Similarity and Dissimilarity Measures in

Clustering Continuous Data. PLoS One *10*, e0144059. https://doi.org/10.1371/journal.pone.0144059.

70. Zhou, Y., and Sharpee, T.O. (2022). Using Global t-SNE to Preserve Intercluster Data Structure. Neural Comput. *34*, 1637–1651. https://doi.org/10.1162/neco_a_01504.
71. Schober, P., Boer, C., and Schwarte, L.A. (2018). Correlation Coefficients: Appropriate Use and Interpretation. Anesth. Analg. *126*, 1763–1768. https://doi.org/10.1213/ANE.0000000000002864.
72. Kobak, D., and Berens, P. (2019). The art of using t-SNE for single-cell transcriptomics. Nat. Commun. *10*, 5416. https://doi.org/10.1038/s41467-019-13056-x.
73. Gove, R., Cadalzo, L., Leiby, N., Singer, J.M., and Zaitzeff, A. (2022). New guidance for using t-SNE: Alternative defaults, hyperparameter selection automation, and comparative evaluation. Visual Informatics *6*, 87–97. https://doi.org/10.1016/j.visinf.2022.04.003.
74. Allen, C., and Hospedales, T. (2019). Analogies Explained: Towards Understanding Word Embeddings. In International Conference on Machine Learning (PMLR), pp. 223–231.

STAR★METHODS

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Deposited Data		
Document S1-S15	This manuscript	https://doi.org/10.5281/zenodo.13092715
Software and Algorithms		
FastText	Facebook AI Research	https://fasttext.cc/
Python version 3.11.4	Python Software Foundation	https://www.python.org
Conda version 23.7.2	Anaconda, Inc.	https://www.anaconda.com/
ChemDataExtractor	chemdataextractor.org[65]	http://chemdataextractor.org
t-SNE implementation	scikit-learn[66]	https://scikit-learn.org/
pdfplumber	Python library	https://github.com/jsvine/pdfplumber
icu_tokenizer	ICU project	https://github.com/unicode-org/icu
Wordninja	Python library	https://github.com/keredson/wordninja
GPT-ada-002	OpenAI	Accessed via OpenAI API
BERT-base-uncased	Google AI	Accessed via TensorFlow/Transformers libraries
Complete codebase for this manuscript	Developed in-house	https://doi.org/10.5281/zenodo.13094694
Other		
AMD Ryzen™ 9 7950X3D	Advanced Micro Devices, Inc.	https://www.amd.com/en/products/cpu/amd-ryzen-9-7950x3d

METHOD DETAILS

The methodology which comprises (a) Data Collection, (b) Data Preprocessing, (c) Model Training, and (d) Evaluation Metrics, is illustrated in Figure 1. All models and data processing were trained and executed on a server, fortified with an AMD Ryzen 9 7950X3D 16-core CPU (32 threads) and 128 GB of DDR5 ECC RAM. The versions of Python and Conda used were 3.11.4 and 23.7.2 respectively. The versions of the other required libraries are listed in Data S1 (requirements.txt).

FastText offers two distinct methodologies for deriving word representations: the Skipgram and Continuous Bag of Words (CBOW) models, as depicted in Figure 1C. The Skipgram model is designed to predict a target word based on its surrounding words. The model comprises several key components: an input layer, where context words are encoded as one-hot vectors against a vocabulary of predefined size; a projection layer, which transforms these sparse representations into dense, N-dimensional embeddings through multiplication with a weight matrix; and an output layer, utilizing a softmax function to calculate the probability distribution across the vocabulary, with the aim of predicting adjacent target words. The Skipgram model excels in capturing both semantic and syntactic aspects of words, fine-tuning its weights iteratively to increase the prediction accuracy of true target words within a given contextual window.

Conversely, the CBOW model employs a reverse strategy for word representation. It endeavors to predict a target word from a cluster of surrounding context words. This model features a multi-word input layer, with context words encoded as one-hot vectors. These vectors are then combined in the projection layer to produce an averaged embedding, embodying the collective semantic field of the context. The essence of the CBOW model is encapsulated in its output layer, which uses a softmax function to predict the target word, considering the entire vocabulary as potential outputs.

DATA COLLECTION

During the data collection phase, information about Fe, Cu and Synthesis was extracted from the scientific literature. Initially, the Elsevier application programming interface (API) was used to search for articles containing the keywords "Fe" and "Cu" in the title, abstract, or keywords, with the additional requirement that the content contain the keyword "synthesis". The remaining filter criteria were left at their default settings.[67] A total of 11,563 results were generated from this process. Duplicate articles were removed manually and a total of 6,402 articles were selected for data sources, as presented in Data S2, DOIs.txt. 4,071 articles were downloaded and labeled as data source 1. 2,331 articles were accessible in abstract form due to the subscription limitation, and it was labeled as data source 2. Furthermore, 560 open-access journal

articles were obtained using "chemical" and "classical music" as keywords, and were labeled as data source 3 (Data S14) and data source 4 (Data S15) respectively, aiming to cover a broader range of chemical knowledge and noise in the simulated datasets.

DATA PREPROCESSING

pdfplumber library was used to extract text from Data Source 1, which was then merged with the text content from Data Source 2 to obtain the raw text data, in the data preprocessing stage. A series of progressively refined filters were applied in the text preprocessing workflow, as listed below to ensure the quality of the training data. All libraries and tools utilized in this research were listed in the 'data and code availability' section.

(i) The text was processed using Vahe Tshitoyan's materials science text processing tools, which standardized chemical element names. This was to maintain data consistency and reduce the number of unique words. Selective lowercasing, removal of diacritics, and addressed valence states in chemical formulas were performed.
(ii) The icu_tokenizer library was used to tokenize and normalize the text. In addition, due to the variation of PDF versions, the extracted text often encountered issues such as the merging of multiple words and the loss of spaces between them. Therefore, the Wordninja library was used to identify words and separate contiguous words with spaces.
(iii) Sentences were separated by dots, with each line containing one sentence. A minimum of 40 characters is defined as a sentence in this study. If the number of characters in a sentence was less than 40, adjacent sentences were merged until the condition was met. Unrelated sentences, such as those containing "https://" and "http://", were removed.
(iv) Ultimately, a cleaned, comprehensive dataset related to Fe, Cu, and synthesis, labeled as Chem_Dataset (data source 1 and 2), was obtained. It consisted of 758,650 lines of training data, with a total word and character count of 31,673,184 and a unique word and character count of 833,394. Furthermore, a ChemEnhanced_Dataset (data source 1, 2, and 3) containing 823,776 lines of training data, with a total word and character count of 34,183,912 and a unique word and character count of 885,491, was created. Similarly, a NoiseSimulated_Dataset (data source 1, 2, and 4) containing 809,033 lines of training data, with a total word and character count of 33,720,459 and a unique word and character count of 891,987, was obtained.

CONSTRUCTION OF THE VALIDATION SETS

Four datasets were generated using ChatGPT-4 by requesting pairs of synonyms that are strongly/weakly related to chemical raw materials. Manual screening was done by analyzing the relevance of pairs. A strongly chemistry-related positive test dataset (D1), such as ['ethanol', 'methanol'] pair, was created. The list contained 690 pairs of chemical reagents or compounds. A weakly chemically related positive test dataset (D2) was constructed, and it consisted of 165 pairs, such as ['lignite', 'peat']. A strongly chemically related negative test dataset (D3) was formulated, and it consisted of 340 pairs, for instance ['sodium chloride', 'petroleum']. Lastly, a list of 340 pairs, which were weakly chemically related negative test dataset (D4) was created. ['ethanol', 'eagle'] was one of the examples in the D4 list. The complete lists for these datasets can be found in Data S3–S6 (D1.txt, D2.txt, D3.txt, and D4.txt), respectively. Additionally, an analogy reasoning dataset (D5) was created to validate the model's performance. Generated by ChatGPT-4, this dataset includes 113 entries, such as ['water: liquid oxygen: gas'], and is provided in Data S13.

TRAINING AND VALIDATION OF WORD EMBEDDINGS MODELS

First, a pre-trained version of the FastText model was selected in the public domain, namely wiki-news-300d-1M.vec. This 300-dimensional model, referred to as FastText-Base, was trained on Wikipedia 2017, the UMBC webbase corpus, and the statmt.org dataset. It consisted of 16B tokens.

Then, using a grid search strategy with the hyperparameters list in the below table, namely model types, learning rates, epochs, window sizes, word-N-grams and minimum count and the self-developed Chem_Dataset, a total of 96 FastText word models was developed. Subsequently, the D1-D4 validation sets were used to perform synonym tests on the word vector models to quantify the model's understanding of the semantic relationships between words. The performance of the models was evaluated in terms of average cosine similarity, which is a standard metric used to measure the similarity between two vectors, capturing the directional similarity in high-dimensional space.[68]

The formula 1-2 is as follows:

$$\text{Cosine Similarity}\ (A, B) = \frac{A \cdot B}{\|A\|\,\|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \quad \text{(Equation 1)}$$

$$\text{Average Cosine Similarity} = \frac{1}{N}\sum_{i=1}^{N} \text{Cosine Similarity}\ (A_i, B_i) \quad \text{(Equation 2)}$$

where:

A_i and B_i are the individual vector pairs.
n is the dimensionality of the vectors A and B
N is the total number of vector pairs.

Cosine similarity measures the cosine of the angle between two vectors. When two vectors are identical, their cosine similarity is 1. The average of the total individual cosine similarity was calculated by using Equation 2.[69]

The statistical relationship that quantifies the strength and direction of the linear correlation between variables, which are hyperparameters listed in the below table and average cosine similarity were evaluated by Pearson correlation coefficient, Equation 3. Its values vary from −1 to 1, where 1 indicates a complete positive linear relationship, −1 indicates a complete negative linear relationship, and 0 denotes no linear relationship.[70]

$$r = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \overline{y})^2}} \quad \text{(Equation 3)}$$

where:

n represents the sample size.

x_i and y_i are the individual sample points indexed with i.

$\overline{x}$ and $\overline{y}$ represent the means of the samples respectively.

The best-performing model was identified from the highest average cosine similarity value and named as ChemFastText-Opt. The FastText-Base model was then fine-tuned using the recorded optimal model parameters (ChemFastText-Opt) and the scientific literature dataset Chem_Dataset, resulting in the model ChemFastText-Tuned. The embedding model, text-embedding-ada-002 was also adopted in this study. It was accessed via the open API interface provided by OpenAI. It uses cl100k_base as the tokenizer, has a max input token of 8191, and its output dimensions are 1536. This model is codenamed GPT-ada-002. Furthermore, fine-tuned Bert models based on the BERT-base-uncased pre-trained model (Fine-tuned on large text corpora, including BooksCorpus with over 800 million words from various books, and English Wikipedia, featuring around 2.5 billion words) were also trained on three datasets, with the hyperparameters listed in the below table.

Summary of model specifications and training details

Word Embedding Model name	HyperParameters	Values	Training Corpus
FastText-Base	Default	–	Wikipedia 2017, UMBC webbase, statmt.org
ChemFastText-Opt	Model Types	cbow, skipgram	Chem_Dataset
	Learning Rates	1×10^{-2}, 1×10^{-3}, 1×10^{-4}	
	Epochs	30, 60	
	Window Sizes	5, 10	
	Word N-Grams	1, 2	
	Minimum Counts	1, 3	
	Others	Min/Max Character N-Grams Lengths:2/7 Dimensions: 300 Keep the other parameters as default.	
ChemFastText-Tuned	Optimal parameters based on ChemFastText-Opt.	Model Types: cbow Learning Rates: 1×10^{-3} Epochs: 60 Window Sizes: 10 Word N-Grams: 3 Minimum Counts: 2 Min/Max Character N-Grams Lengths:2/7 Dimensions: 300	Fine-tuned on Chem_Dataset/ ChemEnhanced_Dataset/ NoiseSimulated_Dataset based on the FastText-Base model.
GPT-ada-002	Default	–	Multiple Datasets
BERT-Tuned	Learning Rates	1×10^{-3}, 1×10^{-4}, 1×10^{-5}	Fine-tuned on Chem_Dataset/ ChemEnhanced_Dataset/ NoiseSimulated_Dataset based on the BERT-base-uncased model.
	Batch Size	16	
	Epochs	1	
	Optimizer	AdamW	
	Weight Decay	Default	

This table provides an overview of the hyperparameters and training corpus for different word embedding models, including FastText-Base, ChemFastText-Opt, ChemFastText-Tuned, GPT-ada-002, and BERT-Tuned.

t-Distributed stochastic neighbor embedding (t-SNE)

Four datasets, namely reducing agents, chelating agents, copper salts, and iron salts involved in the nanoFeCu synthesis process were prepared according to the recommendation from ChatGPT4 (refer to Data S7–S10). In addition, the names of all chemical reagents in the Chem_Dataset were extracted and presented as Data S11 using the code in the utils folder).

The central principle of t-SNE is to preserve the distances between similar points in both high-dimensional and low-dimensional spaces. This is achieved by minimizing the Kullback-Leibler divergence between the high-dimensional and the corresponding low-dimensional representations. By retaining both local and global structures of the original high-dimensional space, this technique captures multi-scale structures of data and effectively maps them into two or three dimensions. This facilitates a clear visualization of the inherent relationships within the data and enables the visualization of the reduced high-dimensional data.[71] In this study, the t-SNE algorithm was configured as illustrated in Table 2.

In comparison to principal component analysis (PCA), which is effective for identifying the global structure and principal components of data, it often fails to capture the local relationships crucial for understanding complex datasets like those in chemical informatics.[72] Spectral embedding and locally linear embedding, although powerful in preserving local neighborhoods, might not effectively visualize the multi-scale structures inherent in our data. Thus, t-SNE was selected in this study. To visualize the relationships among word vectors, t-SNE was used to perform dimensionality reduction on high-dimensional word vectors.[73] t-SNE dimensionality reduction was applied to map high-dimensional data into two-dimensional space, and visual processing was performed on the list of chemicals (Data S7–S10). This allowed the users to view and analyze the structure and patterns of the data. By comparing and analyzing the distribution and clustering of these reagents in two-dimensional space, a qualitative assessment of the performances of different models is conducted.

Parameter settings for t-SNE implemented (using the Python scikit-learn package)

Parameter	Value
n_components	2
learning_rate	120
early_exaggeration	12
perplexity	15

Parameter values for the t-SNE algorithm used in the study, including n_components, learning_rate, early_exaggeration, and perplexity.All other parameters are retained at their default values.

SYNONYM ANALYSIS

The synonym analysis was conducted to evaluate the ability of FastText-Base, ChemFastText-Opt, ChemFastText-Tuned, and GPT-ada-002 models to identify synonyms and related terms for selected chemical keywords. A set of keywords was chosen, and nearest-neighbor searches were performed with a breadth of k = 250, To narrow down and extract meaningful recommendations from the Chem_Dataset. The identified list of synonym terms was analyzed to assess the models' understanding and correlation with chemical terminologies.

ANALOGY REASONING ANALYSIS

In the analogy reasoning analysis, the models' capabilities to discern complex semantic relationships between terms were evaluated using structured analogy tasks, formatted as 'A is to B as C is to?'. The ChemFastText-Tuned model was specifically assessed for top-50 accuracy, which quantified its ability to accurately predict the target term within the top fifty guesses.[74]