EXTRACTION OF COMPOSITIONS FROM TEXTS OF MATERIAL SCIENCE ARTICLES

Dhruvil Sheth[1], **Kausik Hira**[1], **Mohd Zaki**[2]
Mausam[1], **N. M. Anoop Krishnan**[1,2]
[1]Yardi School of Artificial Intelligence, [2]Department of Civil Engineering
Indian Institute of Technology Delhi
{dhruvilsheth01, kausikhira}@gmail.com, mohdzaki1995
{mausam, krishnan}@iitd.ac.in

ABSTRACT

Materials composition, property, processing, testing and applications are important parts of materials tetrahedron. Due to variations in composition reporting styles in the text of materials science research papers, extracting them becomes a challenging task. To address this challenge, we present an end-to-end pipeline essential for creating and completing the materials science (MatSci) knowledge bases(KBs). The proposed approach involves creating an automated training dataset using distant supervision and rule-based extraction. This dataset was used to train models for identifying sentences (performed well), reporting the composition, and extracting the composition(performed poorly). To improve the performance of the extraction model, two steps were taken: first, generating additional training using GPT-4, and second, classifying the composition reporting styles in text. This dataset was then used to train the FLAN-T5 language model to extract the compositions from the text. We also compared the performance of our approach with GPT-4 and observed that the performance is quite the same for the cases where the compositions are mentioned in the text in a simplified way. For the cases where composition is reported in the form of equations which require solving arithmetic expressions and substitutions, our proposed model has 14.7% better F1-score than GPT-4.

1 INTRODUCTION

In-depth knowledge within the realms of science and engineering is often encapsulated in domain-specific research papers. Extracting information from these scientific articles requires the development of advanced machine learning methods designed to automate this process. These efforts contribute to the creation of extensive domain-specific Knowledge Bases (KBs) (e.g., Ernst et al. (2015)). Such KBs serve multiple purposes, including enhancing information accessibility for domain researchers (Tsatsaronis et al. (2015); Hamon et al. (2017)) and providing critical data for the development of domain-specific machine learning models (Nadkarni et al. (2021)). Additionally, they have the potential to accelerate scientific discoveries (Jain et al. (2013); Ravinder et al. (2021)).

A recent article on challenges in MatSci Information Extraction (IE) reported that 33.21% of compositions were documented in text Hira et al. (2024). Our work extends DiSCoMaTGupta et al. (2023) and provides end-to-end system for extracting compositions from texts. It classifies the sentence into a composition or non-composition, and further into a direct or equational composition, where the direct composition refers to sentences that do not contain chemical compound percentages in the form of equations. The compositions are extracted directly from the sentences for direct composition. Otherwise, it gives an intermediate composition output in terms of equations, which is then substituted with the corresponding variable values to get the compositions. We evaluate this system with LLMs (large language models) to compare our model's performance.

2 RELATED WORK

Recent advancements have seen the emergence of neural models tailored for various natural language processing (NLP) tasks, particularly focusing on information extraction from textual data. In the materials science domain, researchers leverage NLP tools to automate database creation for machine

learning (ML) applications. Notably, the ChemDataExtractor Swain (2016), an NLP pipeline, has been instrumental in constructing databases specific to material science. Domain-centric models like MatSciBERT are pretrained and fine-tuned to facilitate information extraction tasks in this field Gupta et al. (2022). Exploration into information extraction techniques using distant supervision is evident in the works of Mintz et al. (2009). However, when it comes to the material science domain, specific attention is given to numerical data, posing numerous challenges. Madaan et al. (2016) discusses various challenges and methods associated with extracting numerical data in research, particularly emphasizing the material science context.

The recent strides in NLP have given rise to a computational paradigm where large, pre-trained language models (LM) are fine-tuned for domain-specific tasks. A notable contribution is the proposed instruction-based process for data curation in materials science (MatSci-Instruct)Song et al. (2023a). This process is further refined through fine-tuning the LLaMA-based language modelsSong et al. (2023b). However, there is little research on extracting compositions from texts with chemical compounds and their percentages. Our research helps to extract compositions from the text in a distantly supervised manner which is used to train extraction models.

3 DATASET

3.1 RAW DATA COLLECTION

Using a text-mining API els, we obtained 2400 materials science research papers from the Elsevier Science Direct Database, specifically focusing on papers that include compositions listed in the MatSci Database NGF (2019). These papers were parsed with an XML parser to extract sentences from various sections, including table captions and figure captions.

3.2 TRAINING DATASET CREATION METHODS

3.2.1 DISTANT SUPERVISION

Since the locations of compositions within the text of the papers are unknown, we employed a distant supervision approach. Extracted sentences from a given paper were matched with the compositions listed for that paper in the MatSci database. If all the chemical compounds and their constituent percentages in a sentence matched those in the database within a specified tolerance range, the sentence was included as input and the composition as the output for the training set. However, this method missed many text compositions due to several challenges, as mentioned in Hira et al. (2024).

3.2.2 RULE-BASED PARSING

The rule-based parser in Gupta et al. (2023) was improved to increase coverage and handle equational compositions. For instance, for the input text : "The glass composed of $xSiO_2 - (1-x)Na_2O$ where x=0.2.", the parser generates $[\,[('SiO2',\ '(x)/(x+1-x)'), ('Na2O',\ '(1-x)/(x+1-x)')]\,]$, an intermediate output of compositions in terms of variables (x, y, z) for equational compositions. We developed a rule-based parser to fetch the values of the corresponding variables mentioned in the text, and substituted them into the intermediate output to derive the final compositions.

3.2.3 PROMPTING GPT-4

For extracting compositions from text, gpt4-1106 model via OpenAI Python library was used . The temperature was set to 0.0 for reproducibility, and 8-shot prompting A.2 was used.

3.3 DATASET VERSIONS

We retained 1,880 papers for training, 329 for validation, and 191 for testing. Each input in the training data is a sentence, while the output takes the form of: $[[('SiO2', 20.0), ('Na2O', 80.0)],\ [('SiO2', 40.0), ('Na2O', 60.0)]]$. Each list represents a material composition consisting of tuples, with each tuple containing the name and value of a chemical compound. Since a sentence can contain multiple compounds, the output is structured as a list of lists of tuples. In order to achieve optimal extraction performance, we iteratively refined our dataset through four versions. Starting with Dataset V1 and progressively improving through insights from each iteration, we culminated in the satisfactory results obtained with Version 4. The detailed progression and enhancements of each version are described in the following section.

3.3.1 DATASET VERSION 1: UNSPLIT DATA

The first version of dataset was created by combining samples extracted using both distant supervision 3.2.1 and rule-based parsing 3.2.2. A total of 6,691 composition sentences from 1,880 papers for

training dataset, and 1,399 composition sentences from 329 papers for validation set were extracted. We also trained a binary classifier to identify the presence of composition in a sentence. The -ve and +ve samples (sentences having composition) in the training data 3.3.1 are present in ratio 6:1.

3.3.2 Dataset Version 2: Additional data with GPT-4 for Unsplit Data

As the database NGF (2019) includes compositions from a restricted number of papers, and the rule-based parsing is constrained by the limited coverage of human-defined rules, our Dataset V1 3.3.1 is limited. To overcome this, GPT-4-based prompting extraction was employed.

From an additional 2,500 downloaded papers, 26,664 sentences were classified as positive composition sentences using a binary classifier trained with dataset version 1. Ten composition sentences from each paper were randomly selected, resulting in 8,000 sentences. These were then used for in-context prompting with GPT-4. After cleaning, the final dataset comprised 6,138 composition sentences, addressing issues such as empty outputs and incoherent formats.

3.3.3 Dataset Version 3: Splitting Direct and Equational Composition

The model trained on dataset versions 1 or 2 did not effectively substitute the input (x, y, z) context values in equational compositions. As a result, we split the samples into two sets: one with the composition directly in the sentence and the other with the composition presented as an equation. For example: Direct Composition (DC) = "As40Se60", Equational Composition (EC) = "$\mathrm{As}_x\mathrm{Se}_{1-x}$".

Samples were categorized using the rule-based parser 3.2.2, which can extract both Equational Compositions (EC) with (x, y, z) value substitution, and Direct Compositions (DC) where no substitution is needed. Using a flag to determine if substitutions were made during parsing, the samples were separated. This data was used to train a binary classifier for EC vs DC sentence.

Dataset version 4: Additional Data with GPT-4 for EC and DC Split Data

The additional composition sentences that were extracted in dataset V. 2 3.3.2 were classified into EC and DC. A total of 2,683 direct compositions were extracted, of which 2,373 were finalized after post-processing mentioned in A.4. Similarly, 1080 compositions were extracted for EC using GPT-4.

Table 1: No. of sentences in DC and EC for train, dev, and test splits

No. of sentences\Split	Train	Dev	Test
Direct Composition	7516	1039	870
Equational Composition	2824	416	387

These compositions were incorporated into our training set to enhance extraction. While the above method was employed to create training and validation datasets, compositions in the test dataset were manually annotated to ensure accurate performance evaluation. The final distribution of total sentences across the Train, Val, and Test sets is shown in Table 1.

4 Methodology

4.1 Model Overview

We investigated papers where DisCoMAT failed to extract any composition, and in 96% of them, the compositions were found in text. To address this limitation, we propose extending the extraction process to include text-based compositions.

To achieve satisfactory results, we created four dataset versions, discussed in 3.3. Using training dataset v1, we developed a FlanT5-based binary classifier to detect sentences containing compositions. Additionally, we created another FlanT5-based binary classifier to distinguish between compositions mentioned directly in the text and those presented as equations 3.3.3. We trained models for all four dataset versions, fine-tuning T5-base and FlanT5-Large to extract compositions. Separate models were trained for DC and EC.

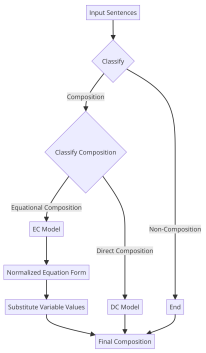

Figure 1: Extraction of compositions from text

Input sentences are classified into composition or non-composition. Composition sentences are further categorized into equational composition (EC) sentences or direct composition (DC) sentences. DC sentences are sent to the DC model, which extracts the final composition directly. EC sentences are handled by the EC model, which extracts composition in a normalized equation form, followed by substituting variable values to obtain the final composition. The entire composition inference is described in Fig. 1.

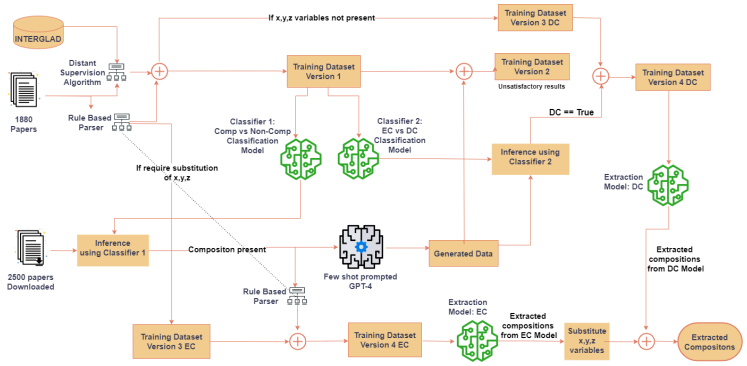

Figure 2: System Architecture with Dataset Versions

Models\Split	VAL	TEST
FlanT5-Large (Trained on v3)	0.819	0.691
FlanT5-Large (Trained on v4)	**0.844**	0.704
GPT-4	0.793	**0.737**
LLaMA 3	0.579	0.516

(a)

Models\Split	VAL	TEST
FlanT5-Large (Trained on v3)	0.691	0.59
FlanT5-Large (Trained on v4)	0.707	**0.618**
GPT-4	0.688	0.471
LLaMA 3	0.418	0.293

(b)

Table 2: (a) End-to-End Comp_Match_Tol scores for DC and (b) for EC

4.2 Evaluation Metric

A correct composition extraction includes accurately extracting all chemical compounds and their constituent percentages within a specified tolerance range, regardless of the order. We introduced a modified F1 score to evaluate this, considering the order invariance of compositions. Precision is calculated by counting matching compositions, while recall counts missed compositions. For example, given a gold list of [[('As', 20.0), ('Se', 58.0), ('Ge', 22.0)], [('As', 20.0), ('Se', 58.0), ('Na', 22.0)]] and a predicted list of [[('Se', 58.0), ('Na', 22.0), ('As', 20.0)]], the second composition matches, resulting in a precision of 1 and a recall of 0.5, yielding an F1 score of 0.67. This metric is referred to as comp_match. To handle compositions within a tolerance range, we allowed a tolerance of 1.0 for composition percentage values, referred to as comp_match_tol.

5 Experiments and Results

5.1 Prompting LLM

Initially, GPT-4's performance was sub-optimal when the output format was a list of lists of tuples. However, we achieved more consistent results and improved scores by converting the output to JSON format. The comp_match and comp_match_tol scores for dataset version 1 are detailed in Table 3. We also evaluated GPT-3.5's performance using OpenAI version gpt-35-turbo-1106 on this dataset. On examining the performance of the LLaMA-3 8B-Instruct Model and GPT-4 on datasets split, while GPT-4 outperformed our model on DC data for test set, it performed poorly on EC data. The LLaMA-3 8B model lagged behind in both DC and EC categories, shown in Table 2

5.2 Model

The F1 scores of the classifiers achieved on the test dataset are 0.99 for Classifier 1 (Comp vs Non-Comp) and 0.98 for Classifier 2 (DC vs EC). The end-to-end extraction scores, after classification into composition and non-composition followed by EC and DC classification, are presented in Table 2. FlanT5-Large was trained on dataset v3 and v4 for both EC and DC.

Our model significantly outperforms GPT-4 on the equational composition dataset, which denotes the lack of mathematical capabilities such as normalization and substitution in LLMs. In contrast, GPT-4 performs slightly better on the direct composition dataset. The comp_match_tol scores for EC in its intermediate form are 0.91 and 0.82, whereas the precision for the final extracted compositions is 0.707 and 0.745 on validation and test dataset respectively, as mentioned in Table. 2.

6 Conclusion and Future Work

In this study, we introduce a novel and challenging task: extracting material compositions from research articles present in the Materials domain. We propose a robust dataset creation pipeline and a baseline system, which classifies the sentences into composition or non-composition sentences and further into direct or equational compositions. This helps handle complex equational compositions and get an intermediate equational composition output, which can be substituted for the corresponding compositions. In future, it will be interesting to use a combination of materials science domain LLMs and the information extraction (IE) pipeline reported in this work to extract material compositions and their properties, which are reported in texts to create a large materials knowledge base.

REFERENCES

Elsevier Developer Portal. URL https://dev.elsevier.com/.

Patrick Ernst, Amy Siu, and Gerhard Weikum. Knowlife: a versatile approach for constructing a large knowledge graph for biomedical sciences. *BMC bioinformatics*, 16:1–13, 2015.

Tanishq Gupta, Mohd Zaki, N. M. Anoop Krishnan, and Mausam. MatSciBERT: A materials domain language model for text mining and information extraction. *npj Computational Materials*, 8(1):102, May 2022. ISSN 2057-3960. doi: 10.1038/s41524-022-00784-w. URL https://www.nature.com/articles/s41524-022-00784-w.

Tanishq Gupta, Mohd Zaki, Devanshi Khatsuriya, Kausik Hira, N M Anoop Krishnan, and Mausam . DiSCoMaT: Distantly supervised composition extraction from tables in materials science articles. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13465–13483, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.753. URL https://aclanthology.org/2023.acl-long.753.

Thierry Hamon, Natalia Grabar, and Fleur Mougin. Querying biomedical linked data with natural language questions. *Semantic Web*, 8(4):581–599, 2017.

Kausik Hira, Mohd Zaki, Dhruvil Sheth, Mausam, and N. M. Anoop Krishnan. Reconstructing the materials tetrahedron: challenges in materials information extraction. *Digital Discovery*, 3(5): 1021–1037, 2024. ISSN 2635-098X. doi: 10.1039/d4dd00032c. URL http://dx.doi.org/10.1039/D4DD00032C.

Anubhav Jain, Shyue Ping Ong, Geoffroy Hautier, Wei Chen, William Davidson Richards, Stephen Dacek, Shreyas Cholia, Dan Gunter, David Skinner, Gerbrand Ceder, et al. Commentary: The materials project: A materials genome approach to accelerating materials innovation. *APL materials*, 1(1), 2013.

Aman Madaan, Ashish Mittal, Mausam, Ganesh Ramakrishnan, and Sunita Sarawagi. Numerical relation extraction with minimal supervision. In *AAAI Conference on Artificial Intelligence*, 2016. URL https://api.semanticscholar.org/CorpusID:2427400.

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. Distant supervision for relation extraction without labeled data. In Keh-Yih Su, Jian Su, Janyce Wiebe, and Haizhou Li (eds.), *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pp. 1003–1011, Suntec, Singapore, August 2009. Association for Computational Linguistics. URL https://aclanthology.org/P09-1113.

Rahul Nadkarni, David Wadden, Iz Beltagy, Noah A Smith, Hannaneh Hajishirzi, and Tom Hope. Scientific language models for biomedical knowledge base completion: an empirical study. *arXiv preprint arXiv:2106.09700*, 2021.

Japan NGF. International glass database system, March 2019. URL https://www.newglass.jp/interglad_n/gaiyo/info_e.html.

Ravinder, Vineeth Venugopal, Suresh Bishnoi, Sourabh Singh, Mohd Zaki, Hargun Singh Grover, Mathieu Bauchy, Manish Agarwal, and NM Anoop Krishnan. Artificial intelligence and machine learning in glass science and technology: 21 challenges for the 21st century. *International journal of applied glass science*, 12(3):277–292, 2021.

Yu Song, Santiago Miret, and Bang Liu. Matsci-nlp: Evaluating scientific language models on materials science language tasks using text-to-schema modeling, 2023a.

Yu Song, Santiago Miret, Huan Zhang, and Bang Liu. Honeybee: Progressive instruction finetuning of large language models for materials science, 2023b.

& Cole J. M. J. Chem. Inf. Model Swain, M. C. Chemdataextractor: A toolkit for automated extraction of chemical information from the scientific literature, 2016.

George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Matthias Zschunke, Michael R Alvers, Dirk Weissenborn, Anastasia Krithara, Sergios Petridis, Dimitris Polychronopoulos, et al. An overview of the bioasq large-scale biomedical semantic indexing and question answering competition. *BMC bioinformatics*, 16:1–28, 2015.

Models\Metric	Comp_Match	Comp_Match_Tol
T5-base	0.568	0.657
FlanT5-Large	0.601	0.681
GPT-4	**0.699**	**0.802**
GPT-3.5	0.578	0.663

Table 3: Comp_Match and Comp_Match_Tol scores for val data of Dataset v1: Unsplitted data

A Appendix

The datasets (automated training data and the manually annotated test data), models, along with their codes used in this study are available at - https://drive.google.com/drive/folders/1GlMckP0IwiInQUqtBvJnZeC13-rAII0a?usp=sharing

A.1 Dataset

A.1.1 Dataset Version 1

The comp_match scores for the model trained are shown in the Table 3. GPT-4 performs well with 8-shot promoting and thus, is used to generate additional training data.

A.2 GPT4-Prompting

For 8-shot prompting of GPT-4 as mentioned in 3.2.3, the prompts are described in the Table: 4.

Role	Content
System	You are a materials science expert. Extract the composition of materials from the following sentence and give output in the fixed JSON format consisting of every composition containing constituent chemical compounds as keys and weight percent as a float value. Make sure you normalize the weights percentages in each list. Do not provide any additional output.
User	Sentence: For making gallium–indium fluoride glass containing 20 mol% GaF_3, 15 mol% InF_3, 30 mol% PbF_2, 20 mol% CdF_2 and 15 mol% ZnF_2, the constituent fluoride compounds in the powder form were mixed inside a dry nitrogen atmosphere glove box.

Assistant	{
	"comp1": {
	"ZnF_2": 15.0,
	"PbF_2": 30.0,
	"GaF_3": 20.0,
	"InF_3": 15.0,
	"CdF_2": 20.0
	}
	}

User	Sentence: 2.1 Preparation of precursor glass Four glasses having composition $(25-x/2)$SrO–$(25-x/2)$CaO–5ZnO–5B_2O_3–40SiO_2–xLa_2O_3 (mol%), where, x=0, 2, 4, 6 were prepared following the conventional melt-quench technique.

Assistant	{
	"comp1": {
	"SiO_2": 40.0,
	"CaO": 25.0,
	"B_2O_3": 5.0,
	"ZnO": 5.0,
	"SrO": 25.0
	},
	"comp2": {
	"SiO_2": 40.0,
	"CaO": 24.0,

	"B_2O_3": 5.0,
	"ZnO": 5.0,
	"SrO": 24.0,
	"La_2O_3": 2.0
	},
	"comp3": {
	"SiO_2": 40.0,
	"CaO": 22.0,
	"B_2O_3": 5.0,
	"ZnO": 5.0,
	"SrO": 22.0,
	"La_2O_3": 6.0
	},
	"comp4": {
	"SiO_2": 40.0,
	"CaO": 23.0,
	"B_2O_3": 5.0,
	"ZnO": 5.0,
	"SrO": 23.0,
	"La_2O_3": 4.0
	}
	}

User	Sentence: The first one was a$R_{12}O(1-a)TeO_2$ where 'a' was 0, 10, 15, 20, 30 mol%, and 'R1' was Li, Na, K. The second one was b$R_{11}O$.c$R_{2\,11\,1}O_3(1-b\text{-}c)TeO_2$ where 'b' was 0, 10, 20, 30 mol%, and 'c' was 0.5% or 16.5%, and 'R_{11}' = Ba, 'R_{111}' = Al, Ga, or In.

Assistant	{
	"comp1": {
	"TeO_2": 100.0
	},
	"comp2": {
	"Li_2O": 20.0,
	"TeO_2": 80.0
	},
	"comp3": {
	"Na_2O": 10.0,
	"TeO_2": 90.0
	},
	"comp4": {
	"Na_2O": 15.0,
	"TeO_2": 85.0
	},
	"comp5": {
	"Na_2O": 20.0,
	"TeO_2": 80.0
	},
	"comp6": {
	"Na_2O": 30.0,
	"TeO_2": 70.0
	},
	"comp7": {
	"K_2O": 20.0,
	"TeO_2": 80.0
	},
	"comp8": {
	"BaO": 20.0,
	"TeO_2": 80.0
	},
	"comp9": {

	"BaO": 16.5, "In_2O_3": 6.5, "TeO_2": 77.0 } }
User	Sentence: The molar composition was the following: 53.3% SiO_2+14.05% B_2O_3+11.3% Na_2O+1.6% ZrO_2+3.4% Al_2O_3+5.0% CaO.

Assistant	{
	"comp1": {
	"SiO_2": 60.12,
	"Na_2O": 12.75,
	"Al_2O_3": 3.84,
	"CaO": 5.64,
	"B_2O_3": 15.85,
	"ZrO_2": 1.8
	}
	}

User	Sentence: The selected compositions are SiBNa404 ($50SiO_2{\cdot}30B_2O_3{\cdot}20Na_2O$, %mol), SiBNa403 ($60SiO_2{\cdot}24B_2O_3{\cdot}16Na_2O$, %mol) both glasses having the same molar ratio B_2O_3/Na_2O=1.5 and commercial Pyrex® glass ($82.8SiO_2{\cdot}10.6B_2O_3{\cdot}3.7Na_2O{\cdot}1.5Al_2O_3{\cdot}1CaO{\cdot}0.3K_2O$, %mol) taken as a reference.

Assistant	{
	"comp1": {
	"SiO_2": 50.0,
	"Na_2O": 20.0,
	"B_2O_3": 30.0
	},
	"comp2": {
	"SiO_2": 60.0,
	"Na_2O": 16.0,
	"B_2O_3": 24.0
	},
	"comp3": {
	"SiO_2": 82.89,
	"Na_2O": 3.7,
	"K_2O": 0.3,
	"Al_2O_3": 1.5,
	"CaO": 1.0,
	"B_2O_3": 10.61
	}
	}

User	Sentence: The $As_{0.4}Se_{0.3}Te_{0.3}$ glass has cross linked As_2Se_3 and As_2Te_3 structural units.

Assistant	{
	"comp1": {
	"As": 40.0,
	"Te": 30.0,
	"Se": 30.0
	}
	}

User	Sentence: All chalcogenide glass membranes $Ag_x(Ge_{0.25}Se_{0.75})_{100-x}$ ($10 \leq x \leq 25$ at.%), $(Ge_{0.25}Se_{0.75})_{75}(Ag_{1-y}Cu_y)_{25}$ (y =0.05, 0.10, 0.20 at.%) and $(Ge_{0.25}Se_{0.75})_{90}(Ag_{0.8}Fe_{0.2})_{10}$ are sensitive to Ag^+, Cu^{2+} and Fe^{3+} ions whereas they do not evidence any response to Mg^{2+} and Cd^{2+}. x=('20', '10', '15', '25', '7'), y=('0.05', '0.2', '0.20', '0.10'), z=('0.4', '0.2')

Assistant	{
	"comp1": {
	"Se": 67.5,

	"Ge": 22.5,
	"Ag": 10.0
	},
	"comp2": {
	"Se": 63.75,
	"Ge": 21.25,
	"Ag": 15.0
	},
	"comp3": {
	"Se": 60.0,
	"Ge": 20.0,
	"Ag": 20.0
	},
	"comp4": {
	"Se": 56.25,
	"Ge": 18.75,
	"Ag": 22.5,
	"Cu": 2.5
	},
	"comp5": {
	"Se": 56.25,
	"Ge": 18.75,
	"Ag": 20.0,
	"Cu": 5.0
	},
	"comp6": {
	"Fe": 2.0,
	"Se": 67.5,
	"Ge": 22.5,
	"Ag": 8.0
	},
	"comp7": {
	"Fe": 4.0,
	"Se": 67.5,
	"Ge": 22.5,
	"Ag": 6.0
	}
	}

User	Sentence: 3.1 Refractive index and material dispersions Fig. 1 shows the refractive index dispersions of (a) SiO_2, 20BaO·80 B_2O_3 (BB), 40BaO·60 SiO_2 (BS), 65CaO·35 Al_2O_3 (CA) and 20Na_2O·80 GeO_2 (NG) glasses and (b) 20Tl_2O·80 TeO_2 (TT), 20Tl_2O·80 Sb_2O_3 (TS) and 80PbO·20 Ga_2O_3 (PG) glasses.

Assistant	{
	"comp1": {
	"SiO_2": 100.0
	},
	"comp2": {
	"B_2O_3": 80.0,
	"BaO": 20.0
	},
	"comp3": {
	"SiO_2": 60.0,
	"BaO": 40.0
	},
	"comp4": {
	"Al_2O_3": 35.0,
	"CaO": 65.0
	},
	"comp5": {

	"Na_2O": 20.0,
	"GeO_2": 80.0
	},
	"comp6": {
	"Tl_2O": 20.0,
	"TeO_2": 80.0
	},
	"comp7": {
	"Sb_2O_3": 80.0,
	"Tl_2O": 20.0
	},
	"comp8": {
	"PbO": 80.0,
	"Ga_2O_3": 20.0
	}
	}

Table 4: GPT-4 prompts for 8-shot prompting

A.3 FLOWCHART FOR COMPOSITION EXTRACTION FROM TEXT

The flowchart for the entire inference process is as explained in the Fig. 1. The process starts with classifying a sentence into composition and non-composition sentence, and again classification of equational and direct composition. These sentences are then passed to respective models to get the final extracted compositions.

A.4 POST-PROCESSING USED FOR REMOVING INVALID COMPOSITIONS

The post-processing method involved eliminating sentences that did not meet the required format, for instance extractions where the total percentage of the constituent compounds in a composition did not equal 100 (allowing for minor tolerance), or where compositions were represented as a single compound (e.g., SiO2, 100.0). Such type of extractions are mostly incorrect, and were therefore removed inorder to enhance the quality of our training data.