

LETTER

# Positioning Political Texts with Large Language Models by Asking and Averaging

Gaël Le Mens<sup>1,2,3</sup>  and Aina Gallego<sup>4,5</sup> 

<sup>1</sup>Department of Economics and Business, Universitat Pompeu Fabra, Barcelona, 08005, Spain; <sup>2</sup>UPF-Barcelona School of Management, Barcelona, 08008, Spain; <sup>3</sup>Barcelona School of Economics, Barcelona, 08005, Spain; <sup>4</sup>Department of Political Science, Constitutional Law and Philosophy of Law, Universitat de Barcelona, Barcelona, 08007, Spain; <sup>5</sup>Institut Barcelona d'Estudis Internacionals, Barcelona, Spain

**Corresponding author:** Gaël Le Mens; Email: [gael.le-mens@upf.edu](mailto:gael.le-mens@upf.edu)

(Received 13 January 2024; revised 5 August 2024; accepted 7 August 2024)

## Abstract

We use instruction-tuned large language models (LLMs) like GPT-4, Llama 3, MiXtral, or Aya to position political texts within policy and ideological spaces. We ask an LLM where a tweet or a sentence of a political text stands on the focal dimension and take the average of the LLM responses to position political actors such as US Senators, or longer texts such as UK party manifestos or EU policy speeches given in 10 different languages. The correlations between the position estimates obtained with the best LLMs and benchmarks based on text coding by experts, crowdworkers, or roll call votes exceed .90. This approach is generally more accurate than the positions obtained with supervised classifiers trained on large amounts of research data. Using instruction-tuned LLMs to position texts in policy and ideological spaces is fast, cost-efficient, reliable, and reproducible (in the case of open LLMs) even if the texts are short and written in different languages. We conclude with cautionary notes about the need for empirical validation.

**Keywords:** LLM; ideology; scaling; text as data

**Edited by:** Jeff Gill

## 1. Introduction

Much research in the social and political sciences involves estimating the positions of actors, such as politicians or political parties, in latent ideological and policy spaces, such as the left-right or liberal-to-conservative continuum. Widely used approaches involve the automatic processing of text documents produced by these actors, such as party manifestos (Laver, Benoit, and Garry 2003; Slapin and Proksch 2008) or legislative speeches (Lauderdale and Herzog 2016). Other approaches involve human coding by experts (Budge 2001) or crowd workers (Benoit *et al.* 2016). Yet other approaches rely on other inputs, such as roll call votes (Poole and Rosenthal 1985), Twitter connections (Barberá 2015), or campaign donations (Bonica 2014).

We propose a new approach to position text documents in ideological and policy spaces using instruction-tuned large language models (LLMs) and evaluate its performance. These models are LLMs optimized for dialog use cases and are typically interacted with *via* chatbots such as ChatGPT. We build on the direct query method introduced by Le Mens *et al.* (2023a) who measured the typicality of text documents in concepts by asking GPT-4 for typicality scores. We directly ask an LLM where a tweet or a sentence of a political text stands on the focal dimension and take the average of the LLM responses to obtain position estimates of longer texts or political actors.

We focus on four scaling tasks using texts of different types, contexts, and lengths. First, we position individual tweets published by US Representatives and Senators by directly asking LLMs where these

stand on the left-right ideological spectrum. Second, we position senators of the 117th US Congress on the same dimension by averaging the position estimates of a sample of the tweets they published during the Congress session. Third, we position party manifestos on the economic and social policy dimensions. We ask the LLMs for the positions of each sentence on these dimensions and average the LLM responses to obtain position estimates of the party manifestos. Fourth, we apply this approach to position speeches by EU legislators, in 10 different languages, about a policy proposal on the ‘anti-subsidy’ to ‘pro-subsidy’ scale. This allows us to explore the potential of this approach for comparative research with multilingual data.

Several articles have shown that LLMs can produce text *annotations* (classifications in discrete categories such as relevant/irrelevant or a topic among a limited set of candidate topics) that are in good agreement with those produced by human coders (e.g., Gilardi, Alizadeh, and Kubli 2023; Törnberg 2025; Ziems *et al.* 2023). However, little work has used LLMs to produce position estimates of political texts in ideological and policy spaces, which is the essence of text *scaling*, a core task in political science (see Benoit *et al.* (2020) for a discussion of the difference between the two tasks). We know of only two recent studies that rely on the text generation capabilities of LLMs to estimate policy and ideological positions. Our approach differs from both. The first study used GPT-3 as a probabilistic text classifier to obtain the posterior probability that a sentence in a party manifesto is “Conservative” or “Liberal” and defined the position of the manifesto as the average (across sentences) of the difference between these probabilities (Ornstein, Blasingame, and Truscott 2024). Our approach directly asks the LLM for the position of text on the focal dimension, and we find that it performs better.<sup>1</sup> The second study asked GPT 3.5 and Llama 2 to compare pairs of politicians on a particular dimension and used these pairwise comparisons to construct estimates of politician position on unidimensional scales (e.g., gun control support) (Wu *et al.* 2023). We ask LLMs to position political *texts* produced by political actors instead of asking them to position political actors based on their names. Our approach is thus applicable even to political actors about whom the LLM has little information.

## 2. Methods and data

### 2.1. Obtaining position estimates with LLMs

Table 1 lists the LLMs we used for text scaling, including their open or closed status. These consist of a set of the most recent and largest LLMs available at the end of May 2024.<sup>2</sup>

**Table 1.** LLMs used for the comparative analyses.

LLM	LLM (full name)	Execution		End of training period	Publisher	Languages
		Open / Closed	(Local / Cloud / API)			
GPT-4o	gpt-4o-2024-05-13	Closed	API	Oct 2023	OpenAI	Undisclosed
GPT-4 Turbo	gpt-4-turbo-2024-04-09	Closed	API	Dec 2023		
GPT-4	gpt-4-0613	Closed	API	Sep 2021		
GPT-3.5 Turbo	gpt-3.5-turbo-0125	Closed	API	Sep 2021	Mistral AI	"It is fluent in English, French, Italian, German, and Spanish" <sup>1</sup>
MiXtral 8X22B	open-mixtral-8x22b	Open	API	Apr 2024?		
MiXtral 8X7Bq6	mixtral-8x7b-instruct-v0.1.Q6_K.gguf	Open	Local	Dec 2023?	Meta	Mostly English language, 5% of training data in 30 other languages <sup>2</sup>
Llama 3 70Bq4	Meta-Llama-3-70B-Instruct-Q4_K_M.gguf	Open	Local	Dec 2023		
Llama 3 70B (HF)	Meta-Llama-3-70B-Instruct	Open	Cloud	Dec 2023		
Llama 3 8B	Meta-Llama-3-8B-Instruct	Open	Local	Mar 2023		
Aya 23 35B (HF)	aya-23-35b	Open	Cloud	May 2023	Cohere	23 languages

Notes: 'Open' models can be downloaded on the user's laptop or desktop and run locally if it has enough video memory. 'Local' execution means that the model was run on our laptop (Apple Macbook pro with M1 Max processor with 64GB of RAM) after downloading the model. Models in .gguf format are versions of the model with compressed weights and were run locally with the llama-cpp-python package. Meta Llama 3 8B was downloaded from Huggingface.co and was run locally without any compression with the mix-lm Python package. 'API' execution means that the model was run for a per-token fee on an infrastructure managed by the LLM provider. 'Cloud' means that the model was run without any compression on a dedicated computer in the cloud computer via a Huggingface Inference Endpoint.

<sup>1</sup>: <https://ai.meta.com/blog/meta-llama-3/> (retrieved May 2, 2024)

<sup>2</sup>: <https://mistral.ai/news/mixtral-of-experts/> (retrieved May 2, 2024)

<sup>1</sup>The correlations between the Expert coding estimates and the positions produced by Ornstein *et al.* (2024) are .92 and .8. See Figures 3 and 4 for our results.

<sup>2</sup>See table A1 for analyses with other model such as Mistral, Gemma or Llama 2.

To obtain a position estimate of a text with an LLM, we submitted a prompt that contained a “user message” instructing it to return such an estimate. For example, to locate a tweet on the left-to-right-wing scale, we used:

You will be provided with the text of a tweet published by a member of the US Congress. Where does this text stand on the ‘left’ to ‘right’ wing scale? Provide your response as a score between 0 and 100 where 0 means ‘Extremely left’ and 100 means ‘Extremely right’. If the text does not have political content, set the score to “NA”. You will only respond with a JSON object with the key Score. Do not provide explanations.

<< Text of the tweet >>

In all cases, we set the temperature parameter to 0, to ensure that the LLM would generate its response by selecting the most likely next token, and thus make the LLM responses as deterministic as possible (to ensure replicability). We also set the maximum number of tokens in the response to 20. This parameter does not affect the nature of the message returned by LLMs; it cuts the response down to 20 tokens if the LLM intended to generate a longer response. This ensures speed (token generation tends to be relatively slow) and limits costs (pay-per-use APIs charge per token submitted in the prompt and per token returned in the response). Finally, whenever this option was available, we set the response format to be a JSON object.

To obtain the position of a party manifesto or a policy speech, we proceeded in a similar way with each sentence of the text documents. We then took the average of the positions of the sentences for which the LLM returned a numeric score, mimicking the approach used by Benoit *et al.* (2016) with human coders.

The supplementary material and the replication package available on Code Ocean<sup>3</sup> provide the exact prompts and further details.

## 2.2. Data

### 2.2.1. Tweets Published by US Congress Members After the Training Cut-off of GPT-4

These data allow us to assess the performance of a modern LLM on prediction data we are certain were not part of the LLM pre-training data. We used the 900 tweets originally analyzed in Le Mens *et al.* (2023a). In November 2023, we recruited 597 Prolific participants to each rate 30 tweets by answering the following question: “Where does this text stand on the “left” to “right” wing scale? If the text does not have political content, select “Not Applicable.”” Participants were not given any instructions as to what we meant by “left” or “right.” The crowdsourced position estimate of a tweet is the average of these ratings. All tweets, except one, received at least one position rating (different from “NA”), leading to a test data set of 899 tweets and their crowdsourced position estimates. This measure is highly reliable overall and within-party (Table 2).

### 2.2.2. Senators of the 117th Congress

We obtained the list of senators from VoteView.com, their Twitter usernames, and downloaded the tweets they published during the Congress session through the Twitter API. We used random samples of 100 tweets published by each senator during the 117th Congress session (January 3, 2021 to January 3, 2023). We excluded two senators who published fewer than 100 tweets during the Congress sessions. We use as a benchmark the first dimension Nokken–Poole period-specific DW-NOMINATE score, a well-established position estimate based on senators’ roll-call votes.

<sup>3</sup><https://codeocean.com/capsule/0323087/tree>

**Table 2.** Reliability of the measures based on human ratings used as benchmark for assessing the performance of position estimates produced with LLMs.

<b>US Congress Tweets</b>	Overall	Democratic Tweets	Republican Tweets	
Left to Right	.92	.80	.87	
<b>UK Manifestos</b>	Overall	Labour	Liberal Democrats	Conservatives
Economic Policy	.99	.98	.87	.84
Social Policy	.99	.98	.92	.97
<b>EU Speeches</b>	Overall	For	Against	
Subsidy policy	.98	.94	.93	

The numbers in the table are split-half correlations with Spearman-brown corrections obtained by averaging 1,000 random splits.

### 2.2.3. British Party Manifestos

We obtained the texts, expert coding estimates, and crowd coding estimates from the replication package of Benoit *et al.* (2016). We positioned the 18 British party manifestos on an economic policy dimension (from left- to right-wing) and on a social policy dimension (from conservative to liberal). We used as a benchmark the Expert Coding estimates that were constructed by Benoit *et al.* based on the sentence position estimates provided by a crowd of experts (political scientists). This measure is overall highly reliable and has varying levels of within-party reliability (table 2).

### 2.2.4. Multilingual Setting: EU Policy Speeches in 10 Languages

We also positioned the 36 speeches of a European Parliament debate on a policy proposal concerning state subsidies originally analyzed in Benoit *et al.* on the pro- to anti-subsidy dimension. These were delivered in 10 different languages by speakers who then voted for or against the proposal. Benoit *et al.* (2016) obtained 6 crowdsourced position estimates for each speech from crowdworkers coding the official translations in English, German, Greek, Italian, Polish, and Spanish. We took the simple average of the six crowd-coding estimates as a benchmark. This setting is challenging not only because of its multilingual nature but also because the speeches vary in style (e.g. technical, case-focused, rhetorical) and require knowledge of the debate context to be understood.

## 3. Results

### 3.1. Tweets Published by Members of the US Congress After the Training Cut-off of GPT-4

The LLMs returned “NA” for a subset of tweets, indicating that they judged that these tweets did not have enough political content to return a position estimate (see Supplementary Material for further discussion). The correlations between the position estimates produced by the best LLMs and crowdsourcing are very high, as shown in Figure 1. Position estimates reflect differences between-party and within-party.

To compare these results with those obtained through approaches that do not require the submission of prompts to an LLM, we computed the typicality of each tweet in the Republican and the Democratic parties using probabilistic text classifiers and defined the position of a tweet as the difference between these two typicalities. The training data consist of approximately 1 million tweets published by members of the US Congress during the 116th and 117th Congress sessions.

We used text classifiers based on fine-tuned BERT (the highest performing approach in Le Mens *et al.* (2023b)), fine-tuned GloVe word embeddings and a naive Bayes classifier based on word frequencies

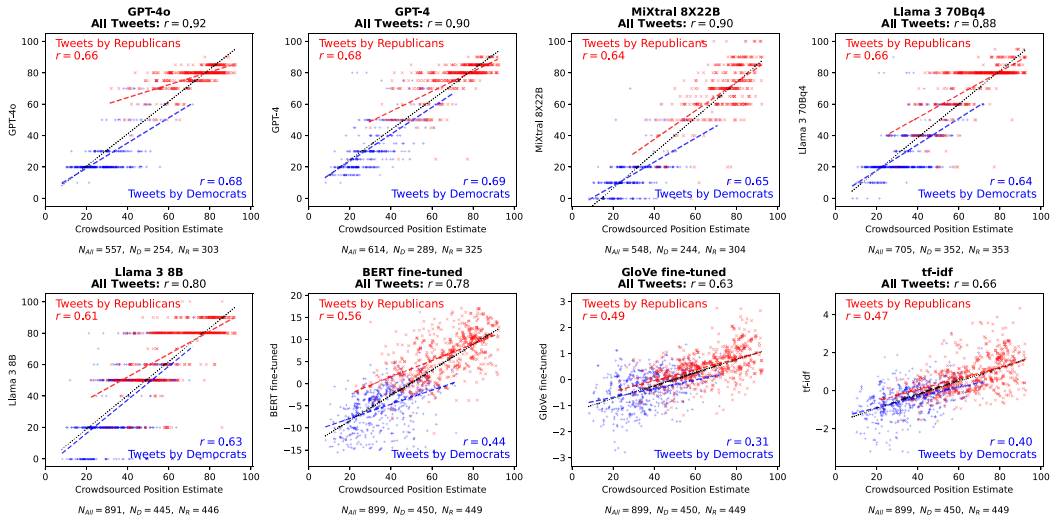


Figure 1. Positioning tweets published by members of the US Congress on the left-right ideological spectrum ( $N = 899$ ).

(TF-IDF).<sup>4</sup> None of these approaches matches the best-performing LLMs, especially when it comes to capturing within-party differences.

### 3.2. Senators of the 117th US Congress

This setting differs from the previous one in that the benchmark positions are not based on human coding but on the voting *behavior* of the senators.

The position estimate of a senator is the average position of their tweets on the left-right ideological spectrum. Figure 2 shows that the resulting position estimates are highly correlated with those based on the roll call votes of the senators during that Congress session (Nokken and Poole 2004), overall and within the party. These correlations are also higher than those produced by supervised classifiers used to obtain typicality measures of tweets in the two parties. The position estimates produced with LLMs are also highly correlated with those based on campaign funding (2020 CF scores, Bonica (2014)), although less so within-party (Figure A8).

### 3.3. British Party Manifestos

The position estimates obtained with the highest performing LLMs are very highly correlated with the Expert Coding estimates, at a level comparable to the position estimates produced with crowd workers (Figures 3 and 4). This is the case not only overall, but also within political parties. These results were obtained without providing any description of the policy dimension to the LLMs. Similar results hold when including such descriptions (Figures A10 and A11).

We also trained a BERT-based supervised probabilistic text classifier (Devlin *et al.* 2018) using the crowdworkers' ratings collected by Benoit *et al.* (2016), and used it to obtain position estimates of the manifestos' sentences and, in turn, of the manifestos. This approach did not yield better results than those obtained with the best LLMs, although the latter were (most likely) not specifically trained to position these party manifestos.

<sup>4</sup>See Le Mens *et al.* (2023a) for an approach that asks LLMs to return typicality ratings.

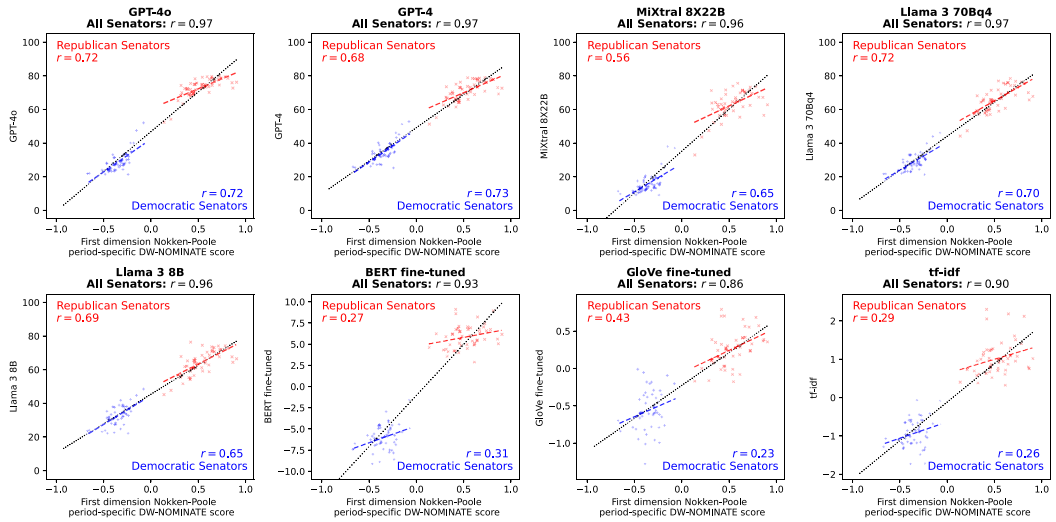


Figure 2. Positioning Senators of the 117th Congress on the left-right ideological spectrum based on a random sample of 100 of their tweets ( $N = 98$ ). Each dot represents a senator ('+' : Democrats, 'x' : Republicans).

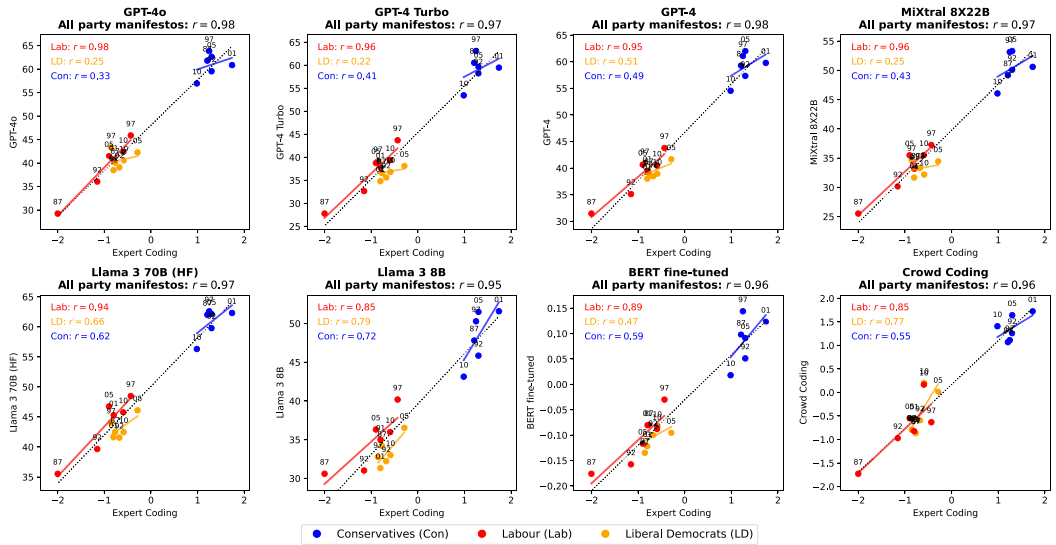


Figure 3. Positioning British party manifestos on the Economic policy dimension (left to right wing scale). The numbers next to the dots indicate the years of the manifestos.

### 3.4. Multilingual Setting: EU Policy Speeches in 10 Languages

We obtained position estimates of the speeches on the “anti-subsidy” to “pro-subsidy” dimension by submitting each sentence to the LLMs *in its original language* with instructions (in English) including background information on the context of the debate.

For the highest performing LLMs (GPT-4o and, to a lesser extent, GPT-4 Turbo, Mixtral 8x22B, Llama 3 70B, Aya 23 35B), the correlation between the benchmark and the position estimates obtained is high overall, and also when we separate the speeches by speakers who voted for and against the policy (Figure 5).

Results obtained with the translations of the speeches in the 6 languages used to obtain the crowd coding estimates show that the best models perform well across languages (Supplementary Material).

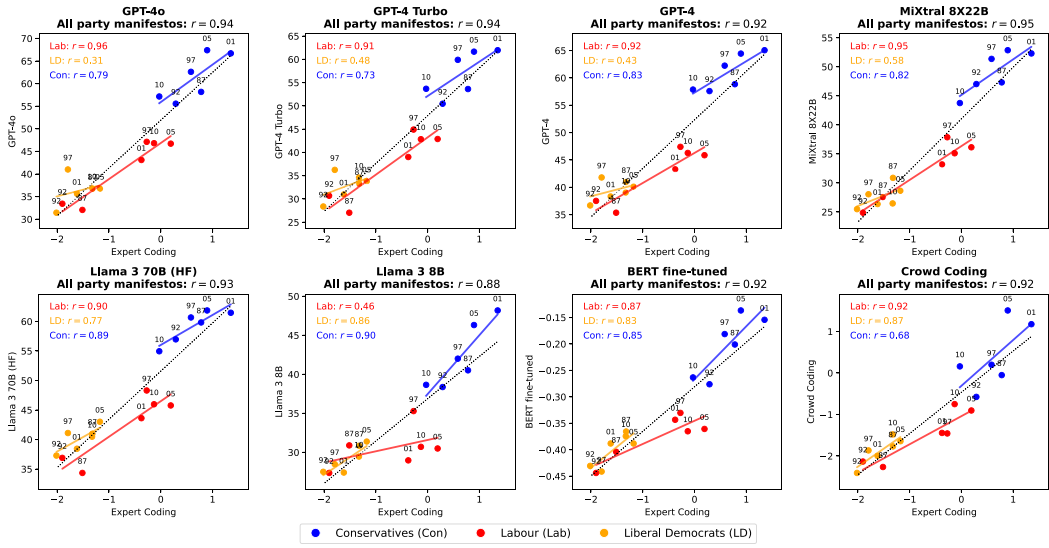


Figure 4. Positioning British party manifestos on the Social policy dimension (liberal to conservative scale). The numbers next to the dots indicate the years of the manifestos.

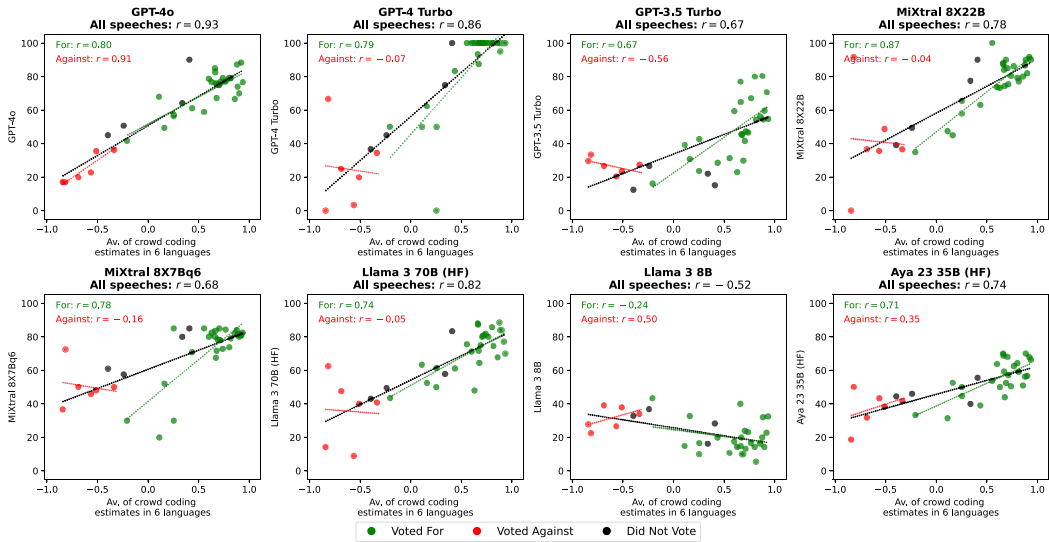


Figure 5. Positioning EU legislative speeches in 10 languages on the “anti-subsidy” to “pro-subsidy” dimension.

#### 4. Discussion

These results demonstrate that “asking” modern instruction-tuned LLMs for the position of short texts in ideological spaces can produce valid position estimates. These can be used, in turn, to position political actors such as politicians, as we illustrated with US senators. We showed that asking LLMs for the positions of sentences in ideological and policy spaces and averaging the responses produces valid positions of party manifestos and policy speeches. The position estimates of the party manifestos produced with the best LLMs are as accurate as the crowdsourced position estimates. And with individual tweets, ancillary analyses show that the position estimates returned by the best LLMs are as accurate as the average of the ratings of about 4 or more independent human coders (Supplementary Material).

This approach has the potential to expand the scope of text analysis due to its high accuracy, speed, ease of implementation, and reproducibility (for open LLMs). Moreover, querying instruction-tuned LLMs is much less costly than human coding even with the most expensive pay-per-use API (GPT-4): \$1.5 versus £1,626 for the 900 tweets analyzed in Section 3.1.

Which LLM should researchers choose? Five main considerations come into play: accuracy with respect to a relevant benchmark, cost, speed, data protection, and reproducibility. If testing reveals no significant accuracy difference between open and closed LLMs, we recommend that researchers use open LLMs such as MiXtral (8X22B) and Llama 3 for text scaling tasks, at least in English. Their architecture and weights are freely available for download, which guarantees the reproducibility of research results and a high level of data protection (if executed locally or on a secure cloud machine).

For multilingual settings, it seems that GPT-4o (a proprietary model) has a marked advantage compared to the best open models at the time of writing, but some LLM designers are releasing open LLMs especially designed to perform well in multiple languages (e.g., the Aya series by Cohere). Some LLMs perform better in some languages than in others (Supplementary Material) and systematic accuracy differences across languages can bias the results of downstream econometric analyses. Developing approaches to deal with this differential measurement error would help realize the potential of LLMs in comparative research.

Another decision is whether to position long text documents in a single prompt or to split them into shorter parts, such as sentences. At this stage, we do not have a clear recommendation on this issue. Positioning party manifestos in a single prompt leads to lower performance than the sentence-by-sentence and averaging approach (Supplementary Material). In contrast, positioning senators by submitting their tweets in a single prompt did not cause significant performance degradation (Supplementary Material). Assessing where and when the single-prompt approach leads to performance degradation is an interesting avenue for future research.

When interpreting the results of the LLM-based approach for positioning political actors such as politicians or parties, it is important to remember that position estimates are based on the text documents submitted to the LLMs. Therefore, the validity of the resulting estimates is limited by the information contained in these texts. In the case of party positions, our approach resembles the approach of the Comparative Manifesto Project but differs from approaches that rely on surveys of experts about their perception of party positions, such as the Chapel Hill Expert Survey. This also implies that the results obtained with LLMs might differ from those obtained from expert surveys in the same way that those obtained by human coding can differ from those obtained with expert surveys because the inputs used to produce the position estimates differ.

The high correlations between position estimates produced with LLMs and human coders reported in this research note could tempt readers to use this approach in other domains while skipping the validation stage. But we advise them *against doing so*. LLMs are well-known for generating biased and unreliable results in some empirical settings. Until other researchers have shown that asking (and averaging) a particular LLM provides accurate scaling results in a variety of empirical settings and focal latent dimensions, we cannot be sure about the breadth of settings in which LLMs perform well for scaling tasks and, *a fortiori*, other measurement or coding tasks. Until more is known about the range of domains in which LLMs perform well at scaling and other measurement tasks, case-by-case empirical validation remains essential.

**Acknowledgments.** We thank the reviewers, Hauke Licht, Christopher Wratil, Xavier Fernandez-Marin, Fabrizio Gilardi, Kenneth Benoit, and participants in the AI-PSR workshop at the University of Barcelona for valuable comments and discussion.

**Data Availability Statement.** Replication code for this article has been published in Code Ocean, a computational reproducibility platform that enables users to run the code, and can be viewed interactively at <https://doi.org/10.24433/CO.0323087.v1>. A preservation copy of the same code and data can also be accessed *via* Dataverse at <https://doi.org/10.7910/DVN/YFM0BW> (Le Mens and Gallego 2024).

**Funding.** This research was funded by ERC Consolidator Grant 772268 from the European Commission to G.L.M, ICREA Academia grants to A.G and G.L.M, grants PID2021-123111OB-I00 (A.G.) and PID2022-137908NB-I00 (G.L.M.) funded by

MICIN/AEI/10.13039/501100011033 and by “ERDF/UE A way of making Europe”, and the Severo Ochoa Programme for Centres of Excellence in R&D (Barcelona School of Economics CEX2019-000915-S) funded by MICIN/AEI/10.13039/501100011033.

**Supplementary Material.** For supplementary material accompanying this paper, please visit <https://doi.org/10.1017/pan.2024.29>.

## References

- Barberá, P. 2015. “Birds of the Same Feather Tweet Together: Bayesian Ideal Point Estimation Using Twitter Data.” *Political Analysis* 23 (1): 76–91.
- Benoit, K., et al. 2020. “Text as Data: An Overview.” In *The SAGE Handbook of Research Methods in Political Science and International Relations*, edited by L. Curini, and R. Franzese, 461–497. London: SAGE Publications Ltd.
- Benoit, K., D. Conway, B. E. Lauderdale, M. Laver, and S. Mikhaylov. 2016. “Crowd-Sourced Text Analysis: Reproducible and Agile Production of Political Data.” *American Political Science Review* 110 (2): 278–295.
- Bonica, A. 2014. “Mapping the Ideological Marketplace.” *American Journal of Political Science* 58 (2): 367–386.
- Budge, I. 2001. *Mapping Policy Preferences: Estimates for Parties, Electors, and Governments*. Vol. 1: 1945–1998. Oxford: Oxford University Press.
- Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova. 2018. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.” arXiv Preprint, [arXiv:1810.04805v2](https://arxiv.org/abs/1810.04805v2).
- Gilardi, F., M. Alizadeh, and M. Kubli. 2023. “Chatgpt Outperforms Crowd Workers for Text-Annotation Tasks.” *Proceedings of the National Academy of Sciences* 120 (30): e2305016120.
- Lauderdale, B. E., and A. Herzog. 2016. “Measuring Political Positions from Legislative Speech.” *Political Analysis* 24 (3): 374–394.
- Laver, M., K. Benoit, and J. Garry. 2003. “Extracting Policy Positions from Political Texts Using Words as Data.” *American Political Science Review* 97 (2): 311–331.
- Le Mens, G., and A. Gallego. 2024. “Replication Data for: Positioning Political Texts with Large Language Models by Asking and Averaging.” Harvard Dataverse, V1. <https://doi.org/10.7910/DVN/YFM0BW>.
- Le Mens, G., B. Kovács, M. T. Hannan, and G. Pros. November 2023a. “Uncovering the Semantics of Concepts Using GPT-4.” *Proceedings of the National Academy of Sciences* 120 (49): e2309350120.
- Le Mens, G., B. Kovács, M. T. Hannan, and G. Pros. March 2023b. “Using Machine Learning to Uncover the Semantics of Concepts: How Well do Typicality Measures Extracted from a Bert Text Classifier Match Human Judgments of Genre Typicality?” *Sociological Science* 10: 82–117.
- Nokken, T. P., and K. T. Poole. 2004. “Congressional Party Defection in American History.” *Legislative Studies Quarterly* 29 (4): 545–568.
- Ornstein, J. T., E. N. Blasingame, and J. S. Truscott. 2024. “How to Train Your Stochastic Parrot: Large Language Models for Political Texts.” Technical report. Working Paper.
- Poole, K. T., and H. Rosenthal. 1985. “A Spatial Model for Legislative Roll Call Analysis.” *American Journal of Political Science* 29 (2): 357–384.
- Slapin, Jonathan B., and S.-O. Proksch. 2008. “A Scaling Model for Estimating Time-Series Party Positions from Texts.” *American Journal of Political Science* 52 (3): 705–722.
- Törnberg, P. 2025. “Large language models outperform expert coders and supervised classifiers at annotating political social media messages.” *Social Science Computer Review*, online first. <https://doi.org/10.1177/08944393241286>.
- Wu, P. Y., J. Nagler, J. A. Tucker, and S. Messing. 2023. “Large Language Models can be Used to Estimate the Latent Positions of Politicians.” Working Paper.
- Ziems, C., W. Held, O. Shaikh, J. Chen, Z. Zhang, and D. Yang. 2023. “Can Large Language Models Transform Computational Social Science?” arXiv Preprint, [arXiv:2305.03514](https://arxiv.org/abs/2305.03514).