大模型论文汇报

王永胜

论文

- STAR: Improving Low-Resource Information Extraction by Structure-to-Text Data Generation with Large Language Models--2023arxiv.org
- Empirical Study of Zero-Shot NER with ChatGPT--2023EMNLP

STAR: Improving Low-Resource Information Extraction by Structure-to-Text Data Generation with Large Language Models

- 解决问题：利用大模型来提高低资源信息抽取的性能
- 创新点：现存低资源IE方法包括：通过迁移学习利用其他任务解决；将任务重新表述为数据丰富的监督任务，严重依赖任务的源数据以及各任务之间的兼容性。作者提出通过大模型合成额外训练数据来微调监督模型的方法STAR（Structure-to-Text DatA GeneRation）
- 方法：1）prompt(定义事件类型)+大模型生成触发词候词。prompt(定义论元角色和实体类型)+大模型生成论元候选词，并创建均匀分布的目标结构。
- 2）通过指令引导文本段落生成。
- 3）通过向推理模型MultiNLI提问问题，不断迭代修正，

实验结果

Figure 1: The STAR inverse data generation strategy using event extraction task as an example. We first generate target structures from valid trigger and argument candidates. Then we prompt the LLM with task instructions from different task granularities to generate the initial passage X_0 containing the event information in the given target structure Y. Finally, we create self-reflection questions to prompt LLM to identify quality issues automatically and refine the passage with template-based hindsight feedback.

EE实验结果

#			Trigger Iden. $k=0$	5	10	Trigger Clas. $k=0$	5	10	Argument Iden. $k=0$	5	10	Argument Clas. $k=0$	5	10
Inference-only Methods														
	LLM	*Formulation*												
1	GPT-3.5	E&IO (Text2Event)	0.00	9.23	11.30	0.00	2.12	3.47	0.00	0.87	1.03	0.00	0.31	0.44
2		E&IO (DEGREE)	0.00	14.39	17.52	0.00	3.17	6.21	0.00	1.02	2.47	0.00	0.92	1.98
3		E&IO (DICE)	0.00	15.13	16.94	0.00	4.11	7.09	0.00	0.71	1.65	0.00	0.33	0.97
4		Task Inst.§	18.31	18.31	18.31	8.37	8.37	8.37		—			—	
5		Inst.+Examples	29.44	47.24	59.71	21.56	40.57	53.29		—			—	
6		Code4Struct		—			—		12.33	18.34	23.74	9.72	14.85	19.10
7	GPT-4	Inst.+Examples	**34.31**	**52.55**	**62.12**	**27.35**	**46.57**	**56.46**		—			—	
8		Code4Struct		—			—		**17.51**	**24.50**	**27.62**	**11.89**	**24.28**	**25.48**
Supervised Models ($N = 50$ except line 9 & 14)														
	EE Model	*Data Creation*												
9	OneIE	None ($N = 0$)	0.00	57.24	60.55	0.00	52.38	54.84	0.00	29.06	36.45	0.00	25.85	33.56
10		Weak Sup.	29.48	49.23	51.66	23.61	45.02	45.23	16.19	24.35	26.84	10.47	19.14	22.94
11		STAR (GPT-3.5)	42.61	63.08	64.12	36.65	56.61	57.29	30.32	39.76	43.40	24.36	36.17	40.93
12		STAR (GPT-4)	**45.42**	**64.63**	**66.77**	**39.15**	**58.84**	**60.76**	**32.23**	**42.76**	**46.22**	**27.47**	**39.53**	**43.25**
13		Human†§	65.62	65.62	65.62	60.10	60.10	60.10	44.76	44.76	44.76	41.60	41.60	41.60
14	DEGREE	None ($N = 0$)	0.00	55.62	57.65	0.00	50.69	52.49	0.00	31.77	42.29	0.00	30.19	40.08
15		Weak Sup.	27.51	46.48	49.70	22.23	41.65	43.55	18.14	32.53	33.33	13.45	27.38	30.01
16		STAR (GPT-3.5)	43.74	61.39	63.57	38.90	56.41	59.10	32.32	48.73	53.06	28.21	46.55	50.97
17		STAR (GPT-4)	**46.69**	**64.47**	**65.17**	**41.75**	**59.92**	**61.42**	**35.85**	**51.92**	**54.56**	**32.09**	**50.74**	**52.99**
18		Human†§	63.49	63.49	63.49	58.86	58.86	58.86	52.47	52.47	52.47	50.09	50.09	50.09

EE实验结果

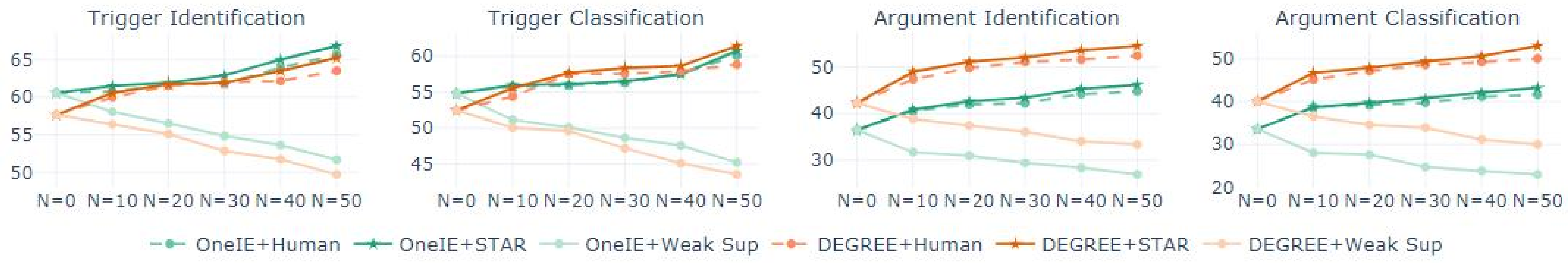

Figure 2: Event extraction performance (F1, %) when the EE models are trained on N augmented training data on top of 10 data points ($k = 10$) for each event type. We observe that performance gain brought by STAR-generated data is magnified as the data augmentation scales up with a larger N, and data generated by STAR is even more effective than human-curated ones. We use GPT-3.5 version STAR for this set of experiments.

RE实验结果

#	RE Model	Data Gen	$N = 0$	10	40
1	GPT-3.5	—	27.91	27.91	27.91
2		Weak Sup.		28.02	28.32
3	SURE	STAR (GPT-3.5)	27.61	**30.50**	**33.02**
4		Human†		30.11	35.62
5		Weak Sup.		30.93	30.29
6	GenPT	STAR (GPT-3.5)	33.38	**34.55**	**37.01**
7		Human†		36.74	37.61

Table 3: Relation extraction performance (%) when the RE models are trained on N augmented training data on top of 10 seed data instances ($k = 10$) for each relation type. We use STAR with GPT-3.5.

Empirical Study of Zero-Shot NER with ChatGPT

- 解决问题：探索大模型在zero-shot命名实体识别任务上的性能。
- 方法：四种策略（decomposed-QA, syntactic prompting, tool augmentation, and two-stage majority voting）

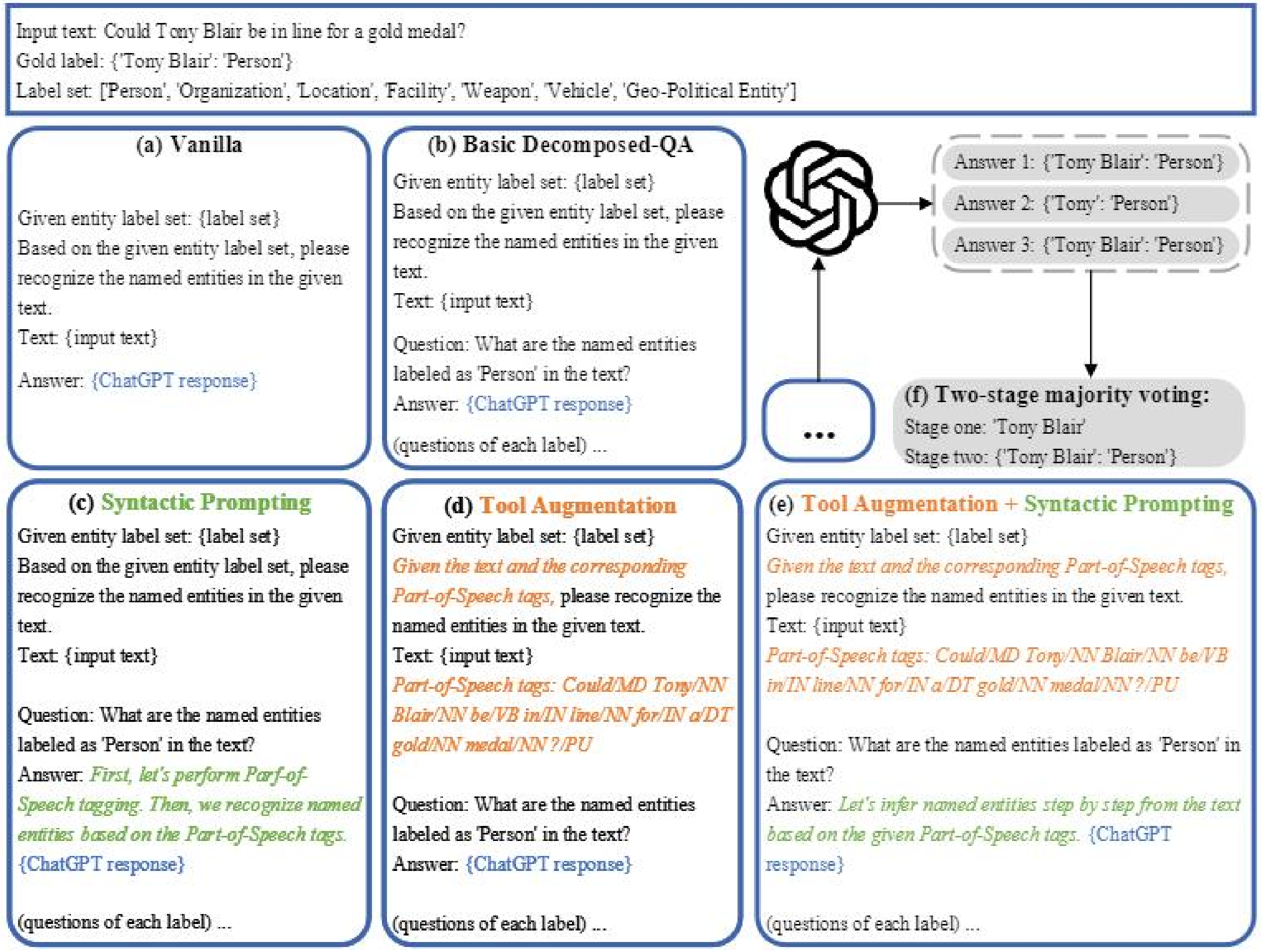

Figure 1: Examples of proposed methods for zero-shot NER with ChatGPT. (a) Vanilla zero-shot method. (b) Basic **decomposed-QA**, where the NER task is broken down into simpler subproblems. (c) Decomposed-QA with **syntactic prompting**. Texts in green are the proposed *syntactic reasoning hint* . (d) Decomposed-QA with **tool augmentation**. Texts in orange are the *content of syntactic information.* (e) Decomposed-QA with tool augmentation and syntactic prompting. (f) SC with **two-stage majority voting**, where stage one votes for the mentions and stage two votes for types. We use part-of-speech tags as an example syntactic information in this figure. The detailed prompts are shown in Appendix H.

Syntactic prompting	
给定实体标签集：[’地缘政治实体’, ’机构名称’, ’地名’, ’人名’]\n 请基于给定的实体标签集，识别给定文本中的命名实体。syntactic reasoning hint (front) \n文本：中国保险监管项目在京启动\n 问题：文本中标签为’人名’的实体有哪些？请以如下JSON格式提供答案：[{’实体名称’: ’实体标签’}]。如果没有对应实体，请返回如下空列表：[]。\n答案：{syntactic reasoning hint (back)} 问题：文本中标签为’地名’的实体有哪些？请以如下JSON格式提供答案：[{’实体名称’: ’实体标签’}]。如果没有对应实体，请返回如下空列表：[]。\n答案：{syntactic reasoning hint (back)} 问题：文本中标签为’机构名称’的实体有哪些？请以如下JSON格式提供答案：[{’实体名称’: ’实体标签’}]。如果没有对应实体，请返回如下空列表：[]。\n答案：{syntactic reasoning hint (back)} 问题：文本中标签为’地缘政治实体’的实体有哪些？请以如下JSON格式提供答案：[{’实体名称’: ’实体标签’}]。如果没有对应实体，请返回如下空列表：[]。\n答案：{syntactic reasoning hint (back)}	
Syntactic reasoning hint (front)	
Word segmentation	首先，你应该进行分词。接着，你应该基于分词结果识别命名实体。
Noun phrases	首先，你应该识别名词。接着，你应该基于名词识别命名实体。
POS tagging	首先，你应该进行词性标注。接着，你应该基于标注的词性识别命名实体。
Constituency parsing	首先，你应该进行成分句法解析。接着，你应该基于成分树识别命名实体。
Dependency parsing	首先，你应该进行依存句法解析。接着，你应该基于依存树识别命名实体。
Syntactic reasoning hint (back)	
Word segmentation	首先，让我们进行分词。接着，我们基于分词结果识别命名实体。
Noun phrases	首先，让我们识别名词。接着，我们基于名词识别命名实体。
POS tagging	首先，让我们进行词性标注。接着，我们基于标注的词性识别命名实体。
Constituency parsing	首先，让我们进行成分句法解析。接着，我们基于成分树识别命名实体。
Dependency parsing	首先，让我们进行依存句法解析。接着，我们基于依存树识别命名实体。

Table 16: Syntactic prompting on Ontonotes 4.

Tool augmentation + syntactic prompting	
给定实体标签集：['地缘政治实体', '机构名称', '地名', '人名']\n{task instruction (involving syntactic tool)}{syntactic reasoning hint (front)}\n文本：中国保险监管项日在京启动\n{syntactic information from tool} 问题：文本中标签为'人名'的实体有哪些？请以如下JSON格式提供答案：[{'实体名称': '实体标签'}]。如果没有对应实体，请返回如下空列表：[]。\n答案：{syntactic reasoning hint (back)} (questions of each label) ...	
{Task instruction (involving syntactic tool)}	
Word segmentation	给定文本和对应的分词结果，请基于实体标签集识别文本中的命名实体。
POS tagging	给定文本和对应的词性标注，请基于实体标签集识别文本中的命名实体。
Constituency parsing	给定文本和对应的成分树，请基于实体标签集识别文本中的命名实体。
Dependency parsing	给定文本和对应的依存树，请基于实体标签集识别文本中的命名实体。
{Syntactic information from tool}	
Word segmentation	分词：['中国', '保险', '监管', '项日', '在', '京', '启动']\n
POS tagging	词性标注：中国/NR 保险/NN 监管/NN 项日/NN 在/P 京/NR 启动/VV\n
Constituency parsing	成分树：(TOP\n (IP\n (NP (NP (NR 中国)) (NP (NN 保险) (NN 监管) (NN 项日)))\n (VP (PP (P 在) (NP (NR 京))) (VP (VV 启动)))))\n
Dependency parsing	依存树：[['中国', '项日', 'nn'], ['保险', '项日', 'nn'], ['监管', '项日', 'nn'], ['项日', '启动', 'nsubj'], ['在', '启动', 'prep'], ['京', '在', 'pobj'], ['启动', '启动', 'root']]\n
{syntactic reasoning hint (front)}	
Word segmentation	请基于给定的分词结果，从文本一步步推理出命名实体。
POS tagging	请基于给定的词性标注，从文本一步步推理出命名实体。
Constituency parsing	请基于给定的成分树，从文本一步步推理出命名实体。
Dependency parsing	请基于给定的依存树，从文本一步步推理出命名实体。
{syntactic reasoning hint (back)}	
Word segmentation	让我们基于给定的分词结果，从文本一步步推理出命名实体。

Word segmentation	请基于给定的分词结果，从文本一步步推理出命名实体。
POS tagging	请基于给定的词性标注，从文本一步步推理出命名实体。
Constituency parsing	请基于给定的成分树，从文本一步步推理出命名实体。
Dependency parsing	请基于给定的依存树，从文本一步步推理出命名实体。
{syntactic reasoning hint (back)}	
Word segmentation	让我们基于给定的分词结果，从文本一步步推理出命名实体。
POS tagging	让我们基于给定的词性标注，从文本一步步推理出命名实体。
Constituency parsing	让我们基于给定的成分树，从文本一步步推理出命名实体。
Dependency parsing	让我们基于给定的依存树，从文本一步步推理出命名实体。

Table 17: Tool augmentation w. / wo. syntactic prompting on Ontonotes 4. If using syntactic prompting, fill in {syntactic reasoning hint}; If not, discard {syntactic reasoning hint}.

实验结果

Method			PPF	PPN	Weibo	MSRA	Onto. 4	ACE05	ACE04
Vanilla			27.85	20.43	30.09	45.51	33.74	28.12	20.09
Decomposed-QA			**36.57**	**30.14**	**34.04**	**48.60**	**37.45**	<u>**34.37**</u>	**22.19**
Syn.	Front	Word segmentation	**38.16**	30.38	32.72	**47.52**	37.47	-	-
		Noun phrases	37.46	30.02	**33.93**	46.05	**38.31**	33.22	20.99
		POS tag	36.89	**30.60**	32.68	46.87	36.82	**34.31**	**21.74**
		Constituency tree	36.21	29.88	31.85	46.02	36.52	33.22	20.86
		Dependency tree	36.33	29.82	33.49	45.61	35.90	34.21	21.04
	Back	Word segmentation	34.89	25.87	32.43	**48.74**	37.48	-	-
		Noun phrases	32.59	24.32	28.71	46.84	38.27	**29.36**	21.74
		POS tag	**36.18**	**26.11**	**33.51**	44.40	36.82	28.84	<u>**23.88**</u>
		Constituency tree	35.71	23.93	30.46	45.84	**39.00**	21.37	18.81
		Dependency tree	31.05	21.02	27.61	44.87	38.52	25.57	21.04
Tool.		Word segmentation	<u>**39.77**</u>	<u>**33.81**</u>	<u>**36.30**</u>	<u>**53.67**</u>	<u>**39.20**</u>	-	-
		POS tag	38.11	30.97	35.14	51.99	37.61	**34.33**	**22.41**
		Constituency tree	36.51	30.25	32.00	48.32	38.40	32.96	22.15
		Dependency tree	39.50	32.12	36.16	48.82	38.05	33.38	22.37
SOTA (fully-supervised)			68.54	70.41	72.77	96.72	84.47	90.90	90.30

Table 1: Overall performance. We report the F1 values. **Vanilla** for vanilla zero-shot method without any techniques; **Syn.** for syntactic prompting; **Tool.** for tool augmentation. We use the same abbreviations in the rest of this paper when necessary. Syntactic augmentation is all conducted under the decomposed-QA setting. Numbers in **bold** are the best results in the corresponding categories; Numbers <u>underlined</u> are the best results among all methods in the zero-shot scenario. The proposed decomposed-QA and syntactic augmentation achieve significant improvements for zero-shot NER on both Chinese and English datasets and on both domain-specific and general-domain scenarios.

Self-Consistency with Two-Stage Majority Voting

- 两阶段：1）若候选词在所有响应中出现的次数过半，则认为该候选词为实体，否则丢弃。
- 2）选择大多数响应的实体标签作为最终实体预测。

- +SC实验结果

Method			PPF	PPN	Onto. 4	ACE05
Vanilla			27.85	20.43	35.16 (1.57)	**29.45** (0.69)
+ SC			**28.85**	**20.72**	**35.79** (1.36)	29.37 (1.35)
Decomposed-QA		-	**36.57**	30.14	38.79 (1.66)	**35.57** (0.83)
+ SC		question-level	33.46	**32.15**	**39.57** (1.50)	31.98 (0.31)
		sample-level	26.98	31.92	39.15 (0.76)	34.38 (0.85)
Syn.	Front	Word segmentation	**38.16**	30.38	37.67 (1.22)	-
		Noun phrases	37.46	30.02	**38.83** (1.24)	34.63 (0.78)
		POS tag	36.89	**30.60**	37.94 (1.49)	34.28 (0.45)
		Constituency tree	36.21	29.88	38.43 (0.84)	34.47 (0.77)
		Dependency tree	36.33	29.82	36.85 (1.16)	**35.77** (0.45)
	Back	Word segmentation	34.89	25.87	39.16 (1.52)	-
		Noun phrases	32.59	24.32	39.52 (0.82)	**29.78** (0.64)
		POS tag	**36.18**	**26.11**	37.00 (2.41)	29.72 (2.06)
		on_conj	35.71	23.93	**40.53** (2.54)	22.23 (0.40)
		Dependency tree	31.05	21.02	39.06 (2.88)	26.65 (0.78)
Syn. + SC	Front	Word segmentation	**38.64**	**32.32**	39.23 (1.13)	-
		Noun phrases	38.16	32.11	**40.34** (1.30)	32.35 (1.18)
		POS tag	38.06	31.75	38.71 (1.91)	33.02 (1.11)
		Constituency tree	37.24	31.60	38.99 (1.52)	32.00 (0.42)
		Dependency tree	37.65	31.30	37.17 (2.21)	**34.59** (0.14)
	Back	Word segmentation	38.43	30.81	40.23 (2.59)	-
		Noun phrases	**38.73**	29.19	39.79 (2.24)	**34.92** (0.72)
		POS tag	38.48	30.77	**40.27** (1.37)	34.40 (1.93)
		Constituency tree	38.02	**31.31**	39.84 (1.90)	33.95 (0.90)
		Dependency tree	37.24	31.20	40.15 (1.94)	34.42 (0.37)
Tool.		Word segmentation	**39.77**	**33.81**	**40.78** (2.58)	-
		POS tag	38.11	30.97	38.15 (2.82)	**35.35** (0.34)
		Constituency tree	36.51	30.25	38.54 (3.19)	34.54 (2.26)
		Dependency tree	39.50	32.12	38.13 (3.04)	34.34 (0.52)
Tool. + SC		Word segmentation	39.63	**33.97**	**41.84** (2.63)	-
		POS tag	37.92	31.72	38.96 (4.21)	33.42 (0.64)
		Constituency tree	36.59	28.35	40.40 (3.98)	**34.60** (0.21)
		Dependency tree	**40.86**	33.59	38.82 (2.61)	30.69 (0.97)
Tool. + Syn.	Front	Word segmentation	**39.67**	**32.97**	**41.09** (3.19)	-
		POS tag	38.85	31.82	39.69 (3.98)	**<u>36.78</u>** (1.36)
		Constituency tree	36.02	30.65	39.44 (2.92)	33.51 (3.04)
		Dependency tree	37.16	32.06	38.83 (3.29)	34.09 (0.78)
	Back	Word segmentation	**36.24**	**31.46**	**39.68** (1.15)	-
		POS tag	34.71	26.51	36.62 (1.05)	**35.70** (1.17)
		Constituency tree	33.76	29.53	39.67 (1.55)	29.64 (2.95)
		Dependency tree	33.18	27.73	36.85 (0.43)	29.19 (2.17)
Tool. + Syn. + SC	Front	Word segmentation	**40.31**	**<u>34.85</u>**	**<u>42.46</u>** (2.20)	-
		POS tag	38.21	30.89	40.86 (2.48)	33.19 (1.39)
		Constituency tree	35.76	29.00	41.36 (3.58)	**33.42** (2.35)
		Dependency tree	39.97	33.23	40.49 (3.49)	30.29 (0.71)
	Back	Word segmentation	40.83	30.78	**41.40** (2.81)	-
		POS tag	38.00	30.64	38.58 (2.77)	**30.28** (2.21)
		Constituency tree	36.26	26.36	40.53 (3.38)	29.78 (1.64)
		Dependency tree	**<u>41.97</u>**	**32.73**	40.19 (2.13)	29.87 (0.17)
SOTA (fully-supervised)			68.54	70.41	84.47	90.90

Table 2: Performance of SC and combinations of reasoning techniques. We report the F1 values. Numbers in parentheses are the standard deviations. Numbers in **bold** are the best results in the corresponding categories; Numbers <u>underlined</u> are the best resul t scenario. SC with two-stage majority voting and combinations of reasoning techniques brings further improvements.

- fewshot上结果

Dataset	Method	0-shot	3-shot	5-shot	10-shot
Ontonotes 4	Vanilla	35.16 (1.57)	38.67 (3.57)	44.51 (5.78)	52.45 (4.13)
	Standard CoT	-	34.34 (6.61)	41.13 (6.31)	41.90 (2.43)
	Tool. w. word segmentation (Ours)	**40.78** (2.58)	42.48 (3.34)	47.16 (5.42)	54.40 (2.68)
	Syn. w. word segmentation (Ours)	37.94 (1.49)	**43.89** (3.67)	**50.70** (7.26)	**56.71** (3.70)
PowerPlantFlat	Vanilla	27.85	35.81 (2.94)	37.44 (3.88)	41.13 (4.89)
	Standard CoT	-	30.63 (6.45)	33.95 (3.59)	38.02 (1.03)
	Tool. w. word segmentation (Ours)	**32.41**	**39.43** (1.91)	**41.12** (4.35)	42.05 (4.74)
	Syn. w. word segmentation (Ours)	28.09	37.84 (2.59)	39.72 (2.79)	**42.52** (3.71)

Table 4: Results under few-shot setting, where the number of shots is the number of texts. We randomly sample three sets of demonstrations and take the averages. Results for Ontonotes 4 are averaged over three sets of randomly sampled 300 samples from the test set. We report F1 values. Numbers in parentheses are the standard deviations. Numbers in **bold** are the best results. Our methods also achieve significant improvements in few-shot scenarios.

Dataset	ACE05			BC5CDR		
Model	GPT-3.5	GPT-3	Llama2	GPT-3.5	GPT-3	Llama2
Vanilla	29.45	14.03	9.07	61.28	29.49	26.12
Decomposed-QA	**35.57**	23.88	15.53	**65.45**	38.73	28.30
Syn. w. dependency tree	26.65	**27.93**	16.98	59.69	41.62	34.46
Tool. w. dependency tree	34.34	27.59	17.31	62.79	**43.69**	**39.94**
Tool. + Syn. w. dependency tree	29.19	18.38	**26.99**	57.28	16.38	39.57

Table 5: Performance on GPT-3 (text-davinci-003) and Llama2 13B chat model. Results are averaged over three sets of randomly sampled 300 samples from the test set. We report the F1 values. Our proposed strategies show consistent improvements on various LLMs.