Annotation Framework for code-mixed Text Classification

Shivasankaran V P
IIT Gandhinagar
vp.shivasan@iitgn.ac.in

Abstract

As the NLP community is shifting gears to solve problems associated with multilingualism, we need robust annotation tools to handle multilingual datasets efficiently. In this paper, we present a code-mixed multilingual text annotation framework, COMMENTATOR, specifically built for code-mixed text. In the current version, we showcase its efficacy in token-level and sentence-level language annotation tasks for the Hinglish text (code-mixing of Hindi and English). We compare COMMENTATOR against a diverse set of five text annotation tools.

1 Exisiting tools

COMMENTATOR tool was installed on Ubuntu 18.04.05 LTS system, see Fig 1. Installation/access of the other five baseline tools can be seen in Fig 2. The full sized screenshots can be found on GDrive[1]

2 Literature Review

The five baseline tools GATE (Cunningham, 2002), UBIAI (ubi), YEDDA (Yang et al., 2017), MARKUP (Dobbie et al., 2021), Tagtog (Cejuela et al., 2014) mentioned in the COMMENTATOR paper bolster many modern Natural language processing(NLP) tasks including text classification, and they work in monolingual settings however they have some challenges. For sentiment analysis, none of the tools provides the annotator with an option to highlight a particular part of the text, which leads the annotator to the final label for the whole text, and this could be crucial information for machine learning models. GATE (Cunningham, 2002), even though it is very modular, the user interface could be overwhelming for an average user. It also has a very limited preset configuration for NLP tasks. Since MARKUP gives active suggestions using the active learning technique to the annotator, there may arise a challenge when the text is displayed in Devanagari script and the model at the backend may not recognize the text, which may create a problem while giving suggestions.

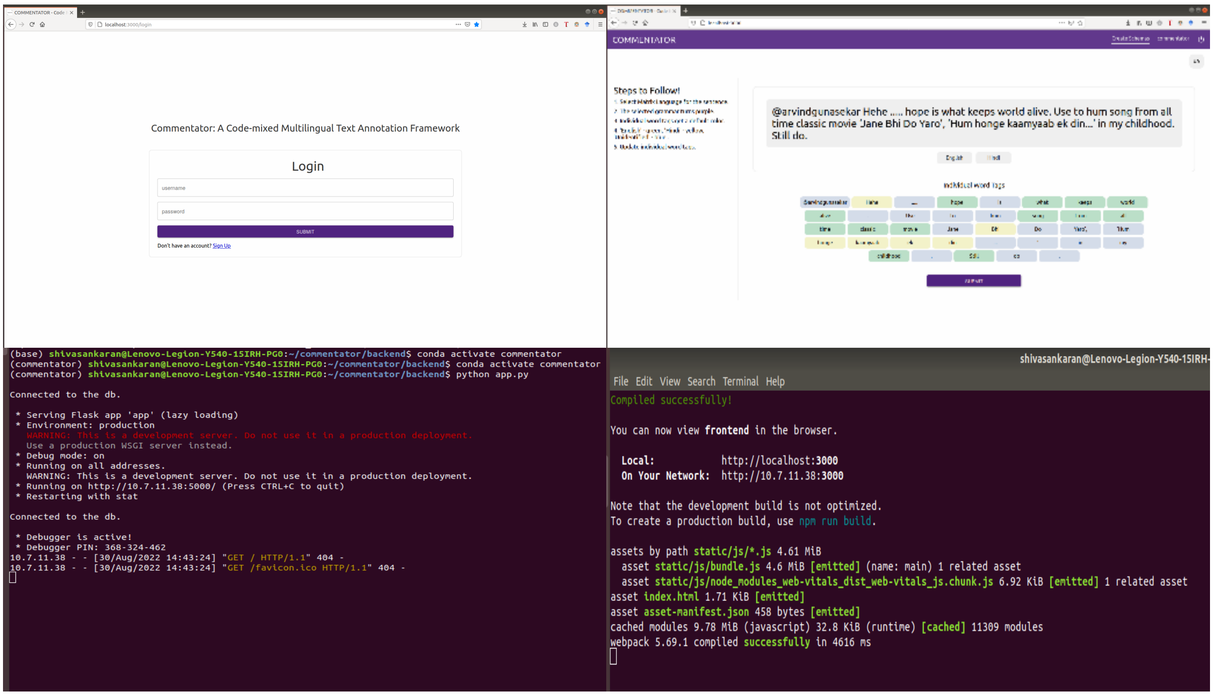
2.1 Other tools

There are ample of Commercial annotation tools for monolingual text annotation tasks like Doccano (Nakayama et al., 2018),Lighttag (Perry, 2021),INCEpTION (Klie et al., 2018),brat (Stenetorp et al., 2012),Prodigy (Pro), all of them support multiple text annotation tasks and bolster a user friendly interface. In contrast, annotation tools specifically designed for code-mixed text classification are limited (Garg and Sharma, 2020). Lighttag (Perry, 2021) supports a hierarchy of annotators for quality control. brat (Stenetorp et al., 2012) leverages a machine learning-based semantic class disambiguation system to decrease the annotation time by 15.4%. Doccano (Nakayama et al., 2018) is an open-source annotation tool which not only supports most of the text annotation tasks but also supports certain multi-modal tasks like image captioning and speech-to-text. INCEpTION (Klie et al., 2018) like brat (Stenetorp et al., 2012) leverages machine learning models to give suggestions to annotators. Works like INCEpTION(Klie et al., 2018), Prodigy (Pro) and (Al-Tamimi et al., 2021) leverages active learning to get the annotations for those texts, which will have maximum impact on a machine learning model.

3 Implementation Plan

We have proposed multiple new features which can be added to the commentator framework, at the right side of the annotator panel, there would be real-time suggestions displayed, this feature was inspired from MARKUP (Dobbie et al., 2021). Also there would be an option for the annotator where

[1] https://drive.google.com/drive/folders/1ExcG34y_ruDLW-2beUvNVuWALvAH5CxM?usp=sharing



Commentator: A Code-mixed Multilingual Text Annotation Framework
Login
username
password
SUBMIT
Don't have an account? Sign Up
COMMENTATOR
Steps to Follow!
English
Hindi
Individual word Tags

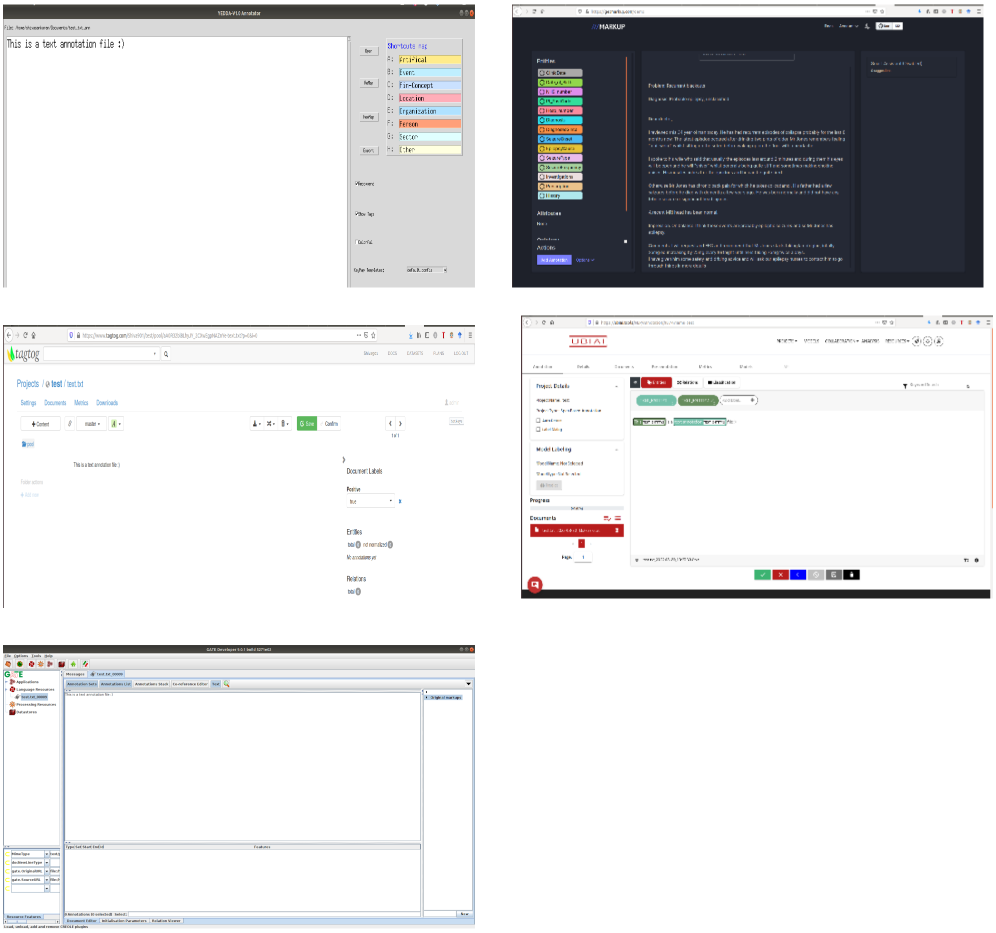
@arvindgunasekar Hehe ..... hope is what keeps world alive. Use to hum song from all time classic movie 'Jane Bhi Do Yaro', 'Hum honge kaamyaab ek din...' in my childhood. Still do.

```
(base) shivasankaran@Lenovo-Legion-Y540-15IRH-PG0:~/commentator/backend$ conda activate commentator
(commentator) shivasankaran@Lenovo-Legion-Y540-15IRH-PG0:~/commentator/backend$ conda activate commentator
(commentator) shivasankaran@Lenovo-Legion-Y540-15IRH-PG0:~/commentator/backend$ python app.py

Connected to the db.

 * Serving Flask app 'app' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Running on all addresses.
   WARNING: This is a development server. Do not use it in a production deployment.
 * Running on http://10.7.11.38:5000/ (Press CTRL+C to quit)
 * Restarting with stat

Connected to the db.

 * Debugger is active!
 * Debugger PIN: 368-324-462
10.7.11.38 - - [30/Aug/2022 14:43:24] "GET / HTTP/1.1" 404 -
10.7.11.38 - - [30/Aug/2022 14:43:24] "GET /favicon.ico HTTP/1.1" 404 -
```

shivasankaran@Lenovo-Legion-Y540-15IRH-
File Edit View Search Terminal Help

```
Compiled successfully!

You can now view frontend in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://10.7.11.38:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

assets by path static/js/*.js 4.61 MiB
  asset static/js/bundle.js 4.6 MiB [emitted] (name: main) 1 related asset
  asset static/js/node_modules_web-vitals_dist_web-vitals_js.chunk.js 6.92 KiB [emitted] 1 related asset
asset index.html 1.71 KiB [emitted]
asset asset-manifest.json 458 bytes [emitted]
cached modules 9.78 MiB (javascript) 32.8 KiB (runtime) [cached] 11309 modules
webpack 5.69.1 compiled successfully in 4616 ms
```

Figure 1: Installation of the COMMENTATOR tool



File: /home/shivasankaran/Documents/test.txt.ann

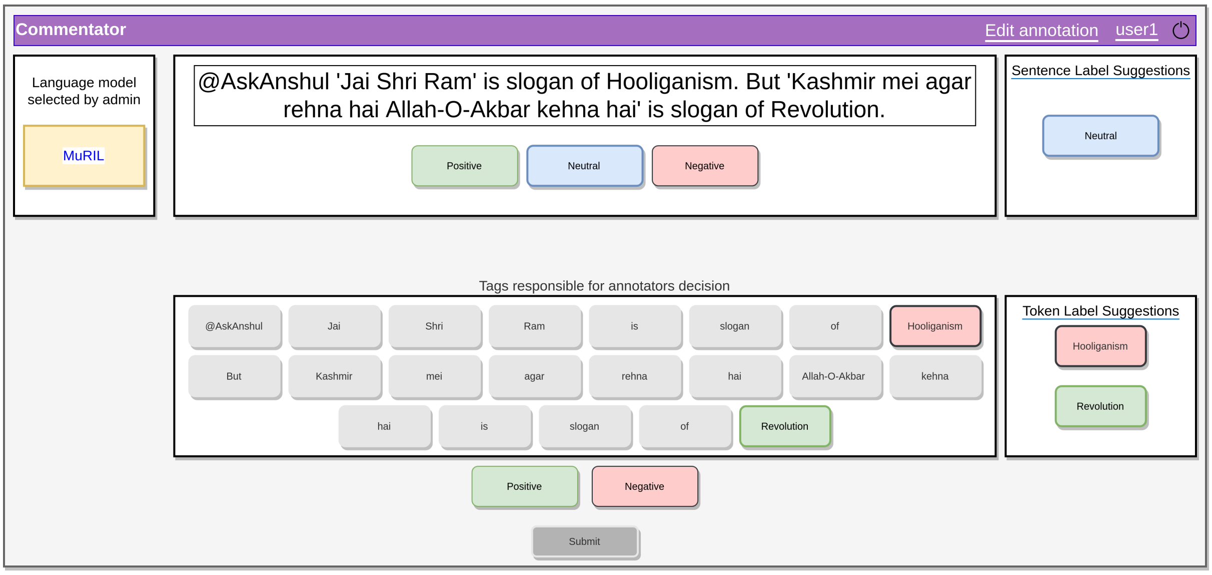
This is a text annotation file :)

Shortcuts map
A: Artifical
B: Event
C: Fin-Concept
D: Location
E: Organization
F: Person
G: Sector
H: Other

Figure 2: Installation/access of YEDDA, MarkUp, TagTog, UBIAI, GATE in the order left to right and top to bottom

Tool Name	**User interface**	**Image support**	**In-text annotation**	**Active Learning**
YEDDA (Yang et al., 2017)	Good	✗	✗	✗
MarkUp (Dobbie et al., 2021)	Good	✗	✓	✓
TagTog (Cejuela et al., 2014)	Good	✓	✗	✓
UBIAI (ubi)	Good	✗	✓	✗
GATE (Cunningham, 2002)	Bad	✗	✗	✗

Table 1: Comparison between different tools.



@AskAnshul 'Jai Shri Ram' is slogan of Hooliganism. But 'Kashmir mei agar rehna hai Allah-O-Akbar kehna hai' is slogan of Revolution.

Figure 3: Proposed interface for annotator's side

he can highlight phrases/words of the displayed sentence which lead him to make his decision of classifying the sentence. At the left side of the panel, an admin can upload and select his own pre-trained/fine-tuned language model. If the language model is capable of handling all the scripts in sentence then appropriate suggestions will be provided to the annotator. In case, the language model is incapable of handling certain scripts which are present in the sentence, we make use of LID tool to differentiate between different scripts, and every other word which the model is incapable of handling and give annotation suggestions based only on the text which the model can handle.

Below are the list of features that can be added to the commentator framework with respect to text classification tasks such as sentiment analysis are as follows.

1. A feature where an annotator can highlight in-text words/phrases which lead to his decision on text. In this case, both full-text and in-text labels will be displayed to the annotator. Specifically, there would be five labels, three for full-text (positive, neutral, and negative) and two for in-text (positive, negative)
2. A feature where a model at the backend makes suggestions by highlighting words/phrases to the annotator. This can be both in-text and full-text words/phrase highlights.
3. A feature where an admin can upload their own pre-trained/fine-tuned models, which would be responsible for making active suggestions displayed on the interface. At the annotator panel, language model name would be displayed.
4. A feature where at the backend, if the text has unrecognized script by the model which would be detected by LID tools, then the model would give suggestions based only on the recognized script.

With the features listed above, for text classification tasks such as sentiment analysis, the commentator framework would solve most of the challenges faced in the existing tools to date.

4 System Description

The modified code can be found here https://github.com/Shiva-sankaran/commentator

4.1 Frontend

A new page has been created for multilingual sentiment analysis task(/sentimental), which closely resembles the LID home page(/) but with 4 tags(positive, negative, neutral, undefined). Further, features are added to the admin page allowing an administrator to upload custom sentiment files to provide word-level sentiment suggestions. Lastly, the administrator can also load a custom model to give sentence-level sentiment suggestion through a hosted huggingface[2] model. Note that it is up to the administrator to make sure that the huggingface model URL given in admin page can handle multilingual texts. A sample URL to handle hinglish sentences can be found here[3].

When an annotator loads a sentence by default the sentence-level and word-level suggestions will be selected. The annotator can then modify the word-level suggestions if necessary by toggling between the 4 possible tags or in the case of sentence-level suggestion the annotator can select the most appropriate sentiment.

4.2 Backend

In order to decrease the annotation time we provide both sentence-level and word-level suggestions. The suggestions are calculated when an administrator uploads a file for the annotators and stored in the database. When an annotator later loads a sentence for annotating the suggestions in the database are gathered along with the sentence and displayed to the annotator.

4.2.1 Sentence-level

A custom sentiment analysis model is loaded from huggingface if the administrator had uploaded an URL to a custom model. The code for obtaining the inference results is made compatible with any general classification model. If no custom model is provided by the administrator then we by default load(Khanuja et al., 2020) to provide suggestions.

4.2.2 Word-level

We process each word indivually to obtain word-level suggestion. If custom word-sentiment files have been uploaded by the administrator, we query the word in the custom files and obtain word-level suggestions. In the case of where no custom word-sentiment files have been given by the administrator we fall back to the established methods of obtaining word-level sentiment as follows.

- For words which belong to English we leverage the lexicon of VADER[4] model in the NLTK(nlt) package to provide word-level suggestions.
- For the words which belong to Hinglish we use the previous work(Yusuf et al., 2022) to convert from Roman script to Devanagari script. Further, we query the converted word in the HindiWordNet(Narayan et al., 2002) for the unique (ID,tag) pair. The (ID,tag) pair can be used to query the sentiment of the word in HindiWordSentiNet(Das and Bandyopadhyay, 2010).

4.3 Database

In order to store sentiment suggestions we create a new collection ’sentiment’. Each entry in the collection consists of the following.

```
tag_id: ’sentence_id’
sentence_tag: ’sentence-level sentiment
word_tags: ’word-level sentiment tags’
```

The ’sentiment’ collection is updated along with the other collections when an administrator uploads a file. The suggestions for the annotator are provided by querying the ’sentiment’ collection.

Collection name	Details
lid	LID Tokens
sentences	Sentences to be annotated
users	Admin and Annotator Accounts
sentiment	Sentence and Word Suggestions

Table 2: Schema of the database

5 Contributions

This was done by one author.

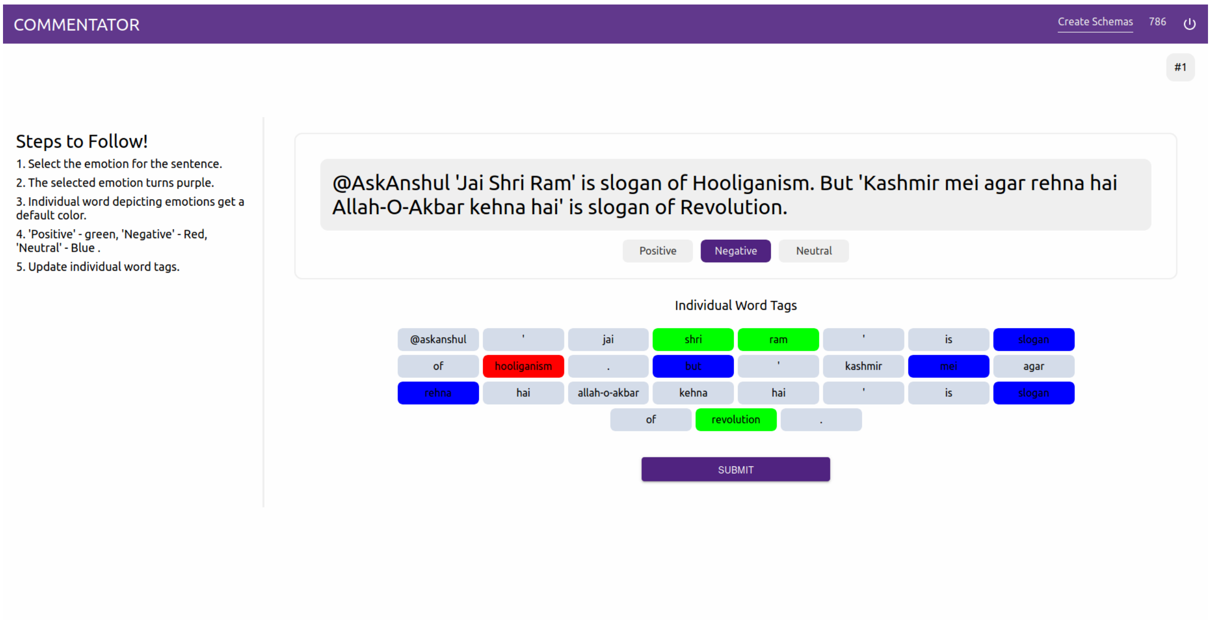
6 Implemented System Images

- Refer to Fig 4 for implemented frontend features
- Refer to Fig 5,6 for change in database after an annotator submits
- Refer to Fig 7,8 for change in user interface when database is manually modified

[2] https://huggingface.co/

[3] https://huggingface.co/ganeshkharad/gk-hinglish-sentiment

[4] https://www.nltk.org/_modules/nltk/sentiment/vader.html

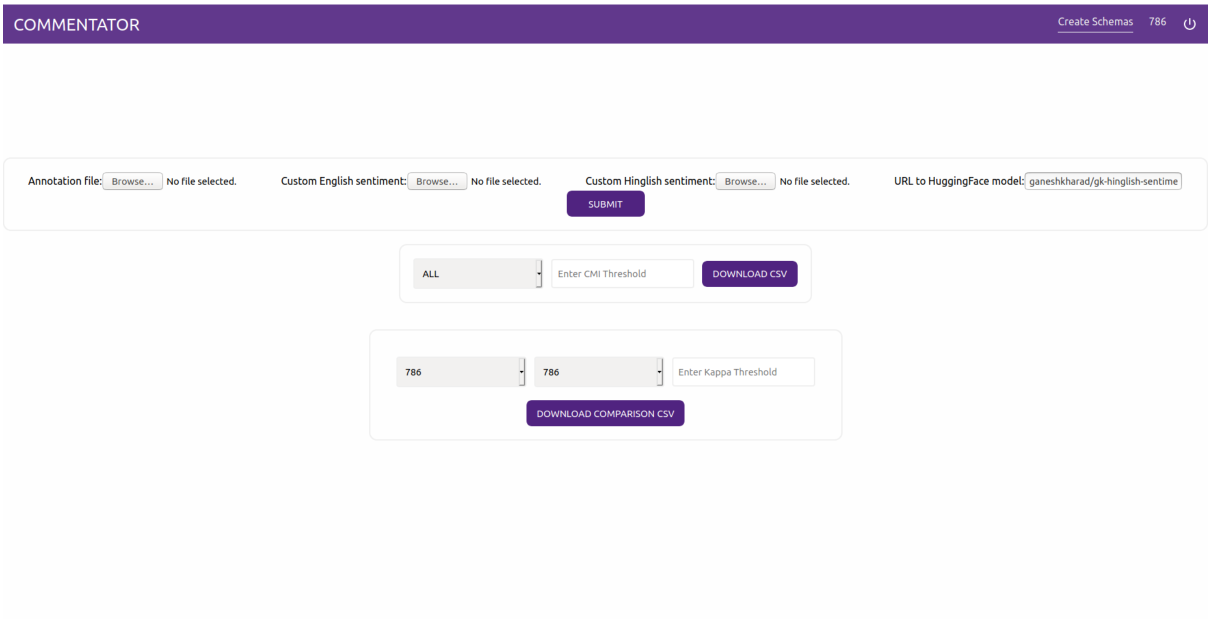


Steps to Follow!

1. Select the emotion for the sentence.
2. The selected emotion turns purple.
3. Individual word depicting emotions get a default color.
4. 'Positive' - green, 'Negative' - Red, 'Neutral' - Blue .
5. Update individual word tags.

@AskAnshul 'Jai Shri Ram' is slogan of Hooliganism. But 'Kashmir mei agar rehna hai Allah-O-Akbar kehna hai' is slogan of Revolution.

(a) Added Sentiment page to the front end



Annotation file: Browse... No file selected.

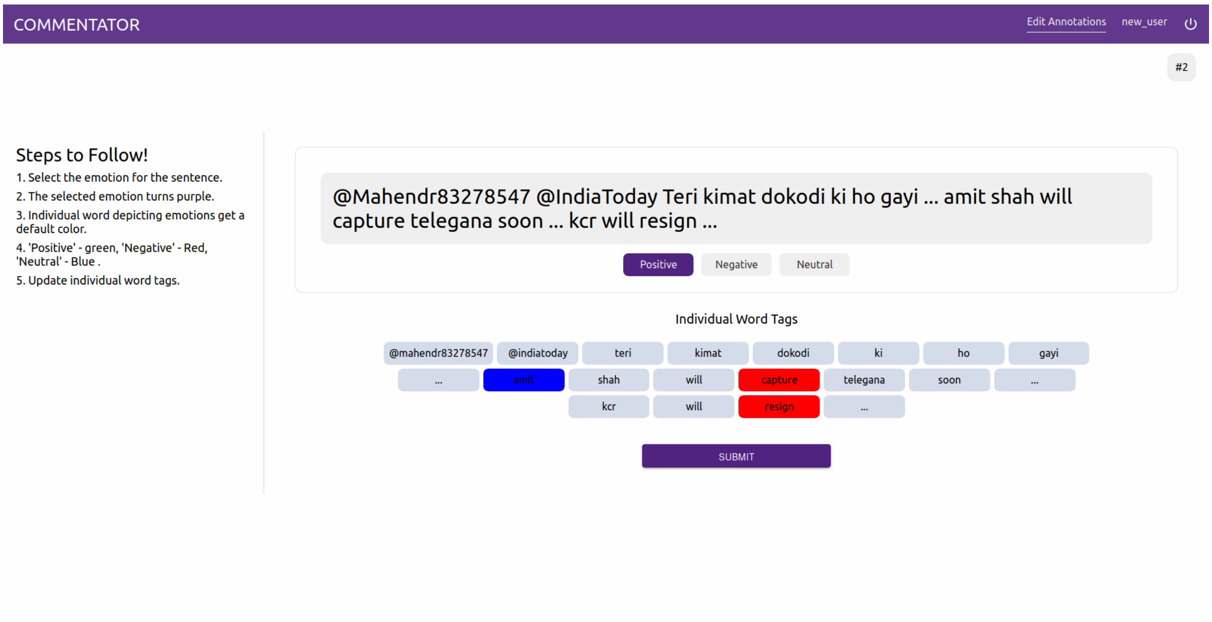
Custom English sentiment: Browse... No file selected.

Custom Hinglish sentiment: Browse... No file selected.

URL to HuggingFace model: ganeshkharad/gk-hinglish-sentime

(b) Added features to upload word-sentiment files and model in the admin page

Figure 4: User interface

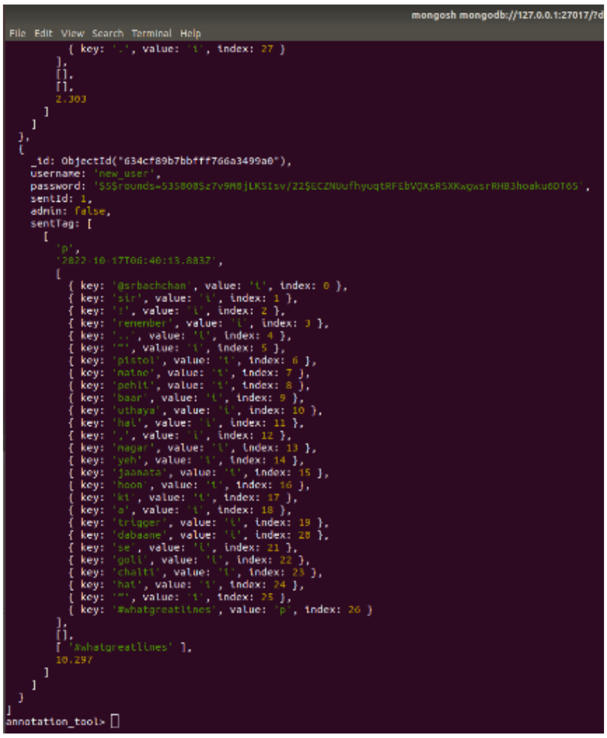


Steps to Follow!

1. Select the emotion for the sentence.
2. The selected emotion turns purple.
3. Individual word depicting emotions get a default color.
4. 'Positive' - green, 'Negative' - Red, 'Neutral' - Blue .
5. Update individual word tags.

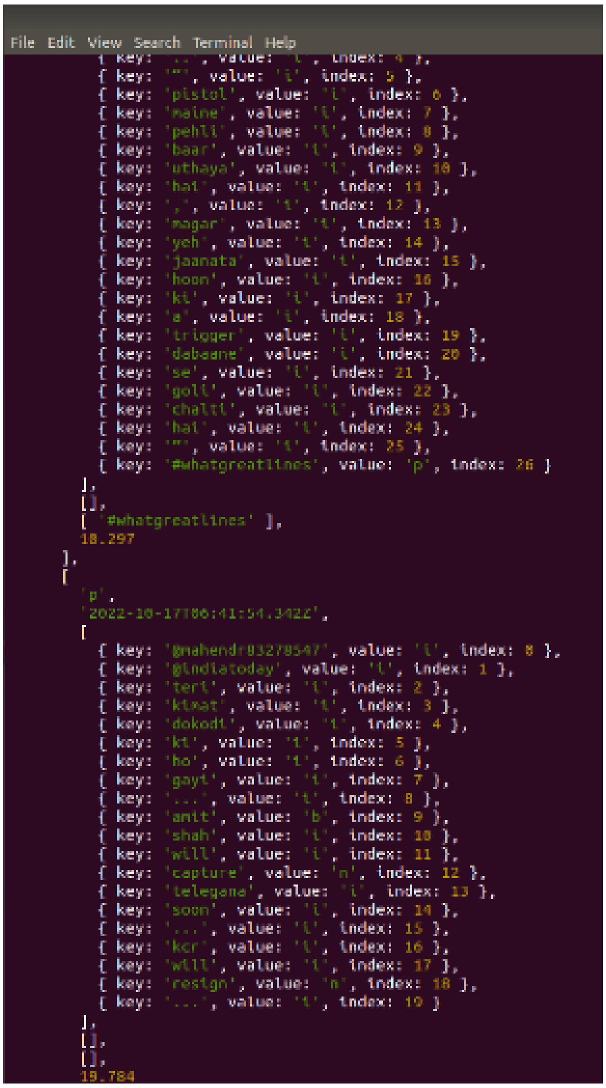
@Mahendr83278547 @IndiaToday Teri kimat dokodi ki ho gayi ... amit shah will capture telegana soon ... kcr will resign ...

Figure 5: Annotator interface: Submitting annotation


mongosh mongodb://127.0.0.1:27017/?d
File Edit View Search Terminal Help

```
          { key: '.', value: 'i', index: 27 }
        ],
        [],
        [],
        2.303
      ]
    ]
  },
  {
    _id: ObjectId("634cf89b7bbfff766a3499a0"),
    username: 'new_user',
    password: '$5$rounds=535808$z7v9M8jLKSIsv/22$ECZNUufhyuqtRFEbVQXsRSXKwgwsrRHB3hoaku6DT65',
    sentId: 1,
    admin: false,
    sentTag: [
      [
        'p',
        '2022-10-17T06:40:13.883Z',
        [
          { key: '@srbachchan', value: 'i', index: 0 },
          { key: 'sir', value: 'i', index: 1 },
          { key: '!', value: 'i', index: 2 },
          { key: 'remember', value: 'i', index: 3 },
          { key: '..', value: 'i', index: 4 },
          { key: '"', value: 'i', index: 5 },
          { key: 'pistol', value: 'i', index: 6 },
          { key: 'maine', value: 'i', index: 7 },
          { key: 'pehli', value: 'i', index: 8 },
          { key: 'baar', value: 'i', index: 9 },
          { key: 'uthaya', value: 'i', index: 10 },
          { key: 'hai', value: 'i', index: 11 },
          { key: ',', value: 'i', index: 12 },
          { key: 'magar', value: 'i', index: 13 },
          { key: 'yeh', value: 'i', index: 14 },
          { key: 'jaanata', value: 'i', index: 15 },
          { key: 'hoon', value: 'i', index: 16 },
          { key: 'ki', value: 'i', index: 17 },
          { key: 'a', value: 'i', index: 18 },
          { key: 'trigger', value: 'i', index: 19 },
          { key: 'dabaane', value: 'i', index: 20 },
          { key: 'se', value: 'i', index: 21 },
          { key: 'goli', value: 'i', index: 22 },
          { key: 'chalti', value: 'i', index: 23 },
          { key: 'hai', value: 'i', index: 24 },
          { key: '"', value: 'i', index: 25 },
          { key: '#whatgreatlines', value: 'p', index: 26 }
        ],
        [],
        [ '#whatgreatlines' ],
        10.297
      ]
    ]
  }
]
annotation_tool> 
```

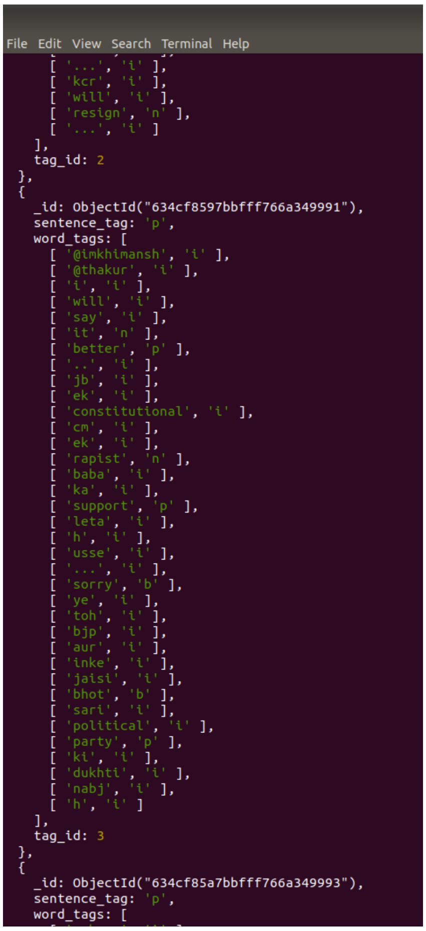
(a) Users collection before annotation submission



```
          { key: '..', value: 'i', index: 4 },
          { key: '"', value: 'i', index: 5 },
          { key: 'pistol', value: 'i', index: 6 },
          { key: 'maine', value: 'i', index: 7 },
          { key: 'pehli', value: 'i', index: 8 },
          { key: 'baar', value: 'i', index: 9 },
          { key: 'uthaya', value: 'i', index: 10 },
          { key: 'hai', value: 'i', index: 11 },
          { key: ',', value: 'i', index: 12 },
          { key: 'magar', value: 'i', index: 13 },
          { key: 'yeh', value: 'i', index: 14 },
          { key: 'jaanata', value: 'i', index: 15 },
          { key: 'hoon', value: 'i', index: 16 },
          { key: 'ki', value: 'i', index: 17 },
          { key: 'a', value: 'i', index: 18 },
          { key: 'trigger', value: 'i', index: 19 },
          { key: 'dabaane', value: 'i', index: 20 },
          { key: 'se', value: 'i', index: 21 },
          { key: 'goli', value: 'i', index: 22 },
          { key: 'chalti', value: 'i', index: 23 },
          { key: 'hai', value: 'i', index: 24 },
          { key: '"', value: 'i', index: 25 },
          { key: '#whatgreatlines', value: 'p', index: 26 }
        ],
        [],
        [ '#whatgreatlines' ],
        10.297
      ],
      [
        'p',
        '2022-10-17T06:41:54.342Z',
        [
          { key: '@mahendr83278547', value: 'i', index: 0 },
          { key: '@indiatoday', value: 'i', index: 1 },
          { key: 'teri', value: 'i', index: 2 },
          { key: 'kimat', value: 'i', index: 3 },
          { key: 'dokodi', value: 'i', index: 4 },
          { key: 'ki', value: 'i', index: 5 },
          { key: 'ho', value: 'i', index: 6 },
          { key: 'gayi', value: 'i', index: 7 },
          { key: '...', value: 'i', index: 8 },
          { key: 'amit', value: 'b', index: 9 },
          { key: 'shah', value: 'i', index: 10 },
          { key: 'will', value: 'i', index: 11 },
          { key: 'capture', value: 'n', index: 12 },
          { key: 'telegana', value: 'i', index: 13 },
          { key: 'soon', value: 'i', index: 14 },
          { key: '...', value: 'i', index: 15 },
          { key: 'kcr', value: 'i', index: 16 },
          { key: 'will', value: 'i', index: 17 },
          { key: 'resign', value: 'n', index: 18 },
          { key: '...', value: 'i', index: 19 }
        ],
        [],
        [],
        19.784
```

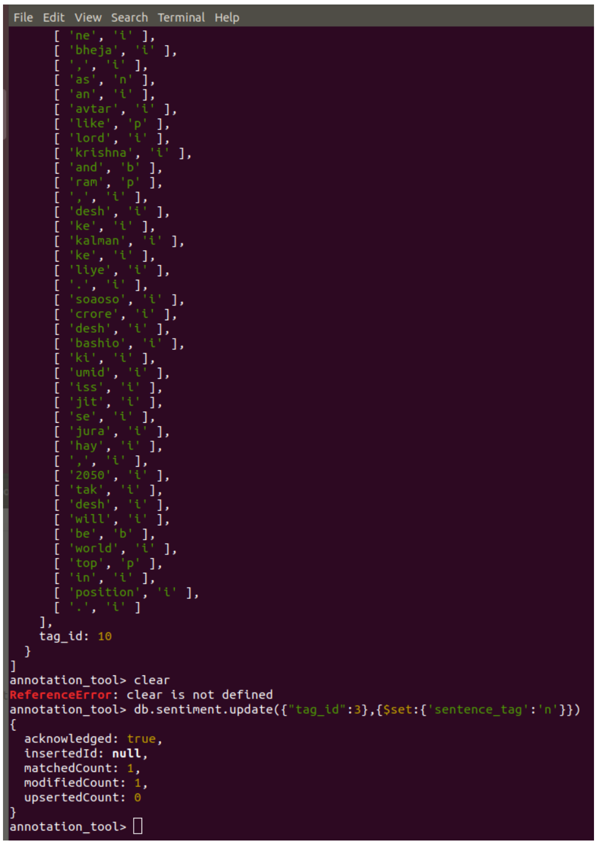
(b) Users collection after annotation submission

Figure 6: Change in database w.r.t change in user interface



```
      [ '...', 'i' ],
      [ 'kcr', 'i' ],
      [ 'will', 'i' ],
      [ 'resign', 'n' ],
      [ '...', 'i' ]
    ],
    tag_id: 2
  },
  {
    _id: ObjectId("634cf8597bbfff766a349991"),
    sentence_tag: 'p',
    word_tags: [
      [ '@imkhimansh', 'i' ],
      [ '@thakur', 'i' ],
      [ 'i', 'i' ],
      [ 'will', 'i' ],
      [ 'say', 'i' ],
      [ 'it', 'n' ],
      [ 'better', 'p' ],
      [ '..', 'i' ],
      [ 'jb', 'i' ],
      [ 'ek', 'i' ],
      [ 'constitutional', 'i' ],
      [ 'cm', 'i' ],
      [ 'ek', 'i' ],
      [ 'rapist', 'n' ],
      [ 'baba', 'i' ],
      [ 'ka', 'i' ],
      [ 'support', 'p' ],
      [ 'leta', 'i' ],
      [ 'h', 'i' ],
      [ 'usse', 'i' ],
      [ '...', 'i' ],
      [ 'sorry', 'b' ],
      [ 'ye', 'i' ],
      [ 'toh', 'i' ],
      [ 'bjp', 'i' ],
      [ 'aur', 'i' ],
      [ 'inke', 'i' ],
      [ 'jaisi', 'i' ],
      [ 'bhot', 'b' ],
      [ 'sari', 'i' ],
      [ 'political', 'i' ],
      [ 'party', 'p' ],
      [ 'ki', 'i' ],
      [ 'dukhti', 'i' ],
      [ 'nabj', 'i' ],
      [ 'h', 'i' ]
    ],
    tag_id: 3
  },
  {
    _id: ObjectId("634cf85a7bbfff766a349993"),
    sentence_tag: 'p',
    word_tags: [
```

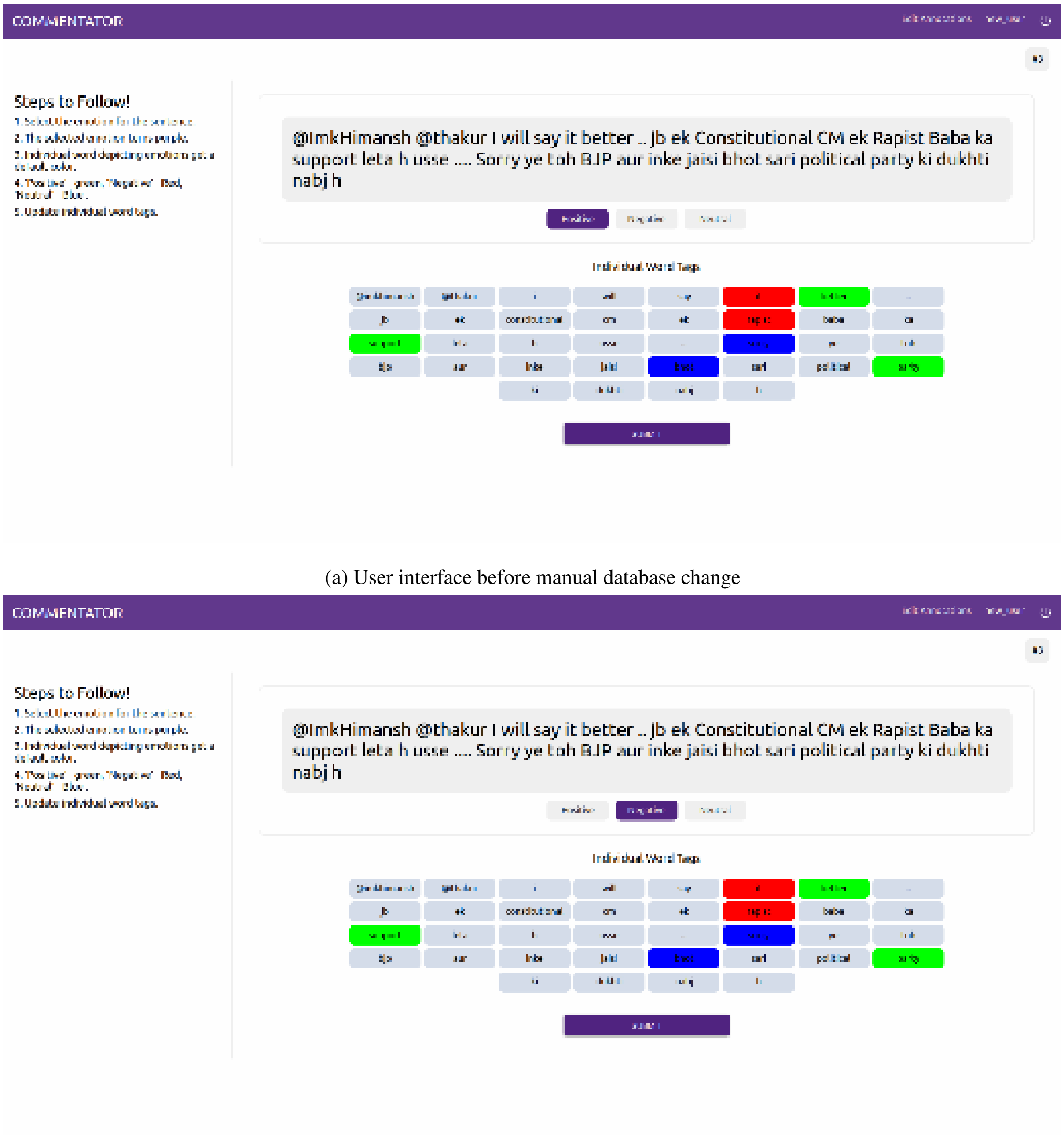
(a) sentiment collection before



```
      [ 'ne', 'i' ],
      [ 'bheja', 'i' ],
      [ ',', 'i' ],
      [ 'as', 'n' ],
      [ 'an', 'i' ],
      [ 'avtar', 'i' ],
      [ 'like', 'p' ],
      [ 'lord', 'i' ],
      [ 'krishna', 'i' ],
      [ 'and', 'b' ],
      [ 'ram', 'p' ],
      [ ',', 'i' ],
      [ 'desh', 'i' ],
      [ 'ke', 'i' ],
      [ 'kalman', 'i' ],
      [ 'ke', 'i' ],
      [ 'liye', 'i' ],
      [ '.', 'i' ],
      [ 'soaoso', 'i' ],
      [ 'crore', 'i' ],
      [ 'desh', 'i' ],
      [ 'bashio', 'i' ],
      [ 'ki', 'i' ],
      [ 'umid', 'i' ],
      [ 'iss', 'i' ],
      [ 'jit', 'i' ],
      [ 'se', 'i' ],
      [ 'jura', 'i' ],
      [ 'hay', 'i' ],
      [ ',', 'i' ],
      [ '2050', 'i' ],
      [ 'tak', 'i' ],
      [ 'desh', 'i' ],
      [ 'will', 'i' ],
      [ 'be', 'b' ],
      [ 'world', 'i' ],
      [ 'top', 'p' ],
      [ 'in', 'i' ],
      [ 'position', 'i' ],
      [ '.', 'i' ]
    ],
    tag_id: 10
  }
]
annotation_tool> clear
ReferenceError: clear is not defined
annotation_tool> db.sentiment.update({"tag_id":3},{$set:{'sentence_tag':'n'}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
annotation_tool> 
```

(b) Manually changing sentiment collection through update command

Figure 7: Changing sentence tag from ’p’ to ’n’ in database manually


COMMENTATOR

edit annotations new_user

Steps to Follow!

1. Select the emotion for the sentence
2. The selected emotion turns purple.
3. Individual word depicting emotions gets a default color.
4. 'Positive' - green, 'Negative' - Red, 'Neutral' - Blue.
5. Update individual word tags.

@lmkHimansh @thakur I will say it better .. Jb ek Constitutional CM ek Rapist Baba ka support leta h usse .... Sorry ye toh BJP aur inke jaisi bhot sari political party ki dukhti nabj h

Positive Negative Neutral

Individual Word Tags

@lmkhimansh @thakur i will say it better ..
jb ek constitutional cm ek rapist baba ka
support leta h usse .... sorry ye toh
bjp aur inke jaisi bhot sari political party
ki dukhti nabj h

Submit

(a) User interface before manual database change

COMMENTATOR

edit annotations new_user

Steps to Follow!

1. Select the emotion for the sentence
2. The selected emotion turns purple.
3. Individual word depicting emotions gets a default color.
4. 'Positive' - green, 'Negative' - Red, 'Neutral' - Blue.
5. Update individual word tags.

@lmkHimansh @thakur I will say it better .. Jb ek Constitutional CM ek Rapist Baba ka support leta h usse .... Sorry ye toh BJP aur inke jaisi bhot sari political party ki dukhti nabj h

Positive Negative Neutral

Individual Word Tags

@lmkhimansh @thakur i will say it better ..
jb ek constitutional cm ek rapist baba ka
support leta h usse .... sorry ye toh
bjp aur inke jaisi bhot sari political party
ki dukhti nabj h

Submit

(b) User interface after manual database change

Figure 8: Change in user interface w.r.t change in database

References

nltk:natural language toolkit.

Prodigy · an annotation tool for ai, machine learning nlp.

Ubiai: Easy to use text annotation tool.

Abdel-Karim Al-Tamimi, Esraa Bani-Isaa, and Ahmed Al-Alami. 2021. Active learning for arabic text classification. In *2021 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE)*, pages 123–126. IEEE.

Juan Miguel Cejuela, Peter McQuilton, Laura Ponting, Steven J Marygold, Raymund Stefancsik, Gillian H Millburn, Burkhard Rost, FlyBase Consortium, et al. 2014. tagtog: interactive and text-mining-assisted annotation of gene mentions in plos full-text articles. *Database*, 2014.

Hamish Cunningham. 2002. Gate, a general architecture for text engineering. *Computers and the Humanities*, 36(2):223–254.

Amitava Das and Sivaji Bandyopadhyay. 2010. Sentiwordnet for indian languages. In *Proceedings of the eighth workshop on Asian language resouces*, pages 56–63.

Samuel Dobbie, Huw Strafford, W Owen Pickrell, Beata Fonferko-Shadrach, Carys Jones, Ashley Akbari, Simon Thompson, and Arron Lacey. 2021. Markup: a web-based annotation tool powered by active learning. *Frontiers in Digital Health*, 3.

N Garg and K Sharma. 2020. Annotated corpus creation for sentiment analysis in code-mixed hindi-english (hinglish) social network data. *Indian Journal of Science and Technology*, 13(40):4216–4224.

Simran Khanuja, Sandipan Dandapat, Anirudh Srinivasan, Sunayana Sitaram, and Monojit Choudhury. 2020. GLUECoS: An evaluation benchmark for code-switched NLP.

Jan-Christoph Klie, Michael Bugert, Beto Boullosa, Richard Eckart de Castilho, and Iryna Gurevych. 2018. The inception platform: Machine-assisted and knowledge-oriented interactive annotation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 5–9. Association for Computational Linguistics. Event Title: The 27th International Conference on Computational Linguistics (COLING 2018).

Hiroki Nakayama, Takahiro Kubo, Junya Kamura, Yasufumi Taniguchi, and Xu Liang. 2018. doccano: Text annotation tool for human. Software available from https://github.com/doccano/doccano.

Dipak Narayan, Debasri Chakrabarti, Prabhakar Pande, and Pushpak Bhattacharyya. 2002. An experience in building the indo wordnet-a wordnet for hindi. In *First international conference on global WordNet, Mysore, India*, volume 24.

Tal Perry. 2021. LightTag: Text annotation platform. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 20–27, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. brat: a web-based tool for NLP-assisted text annotation. In *Proceedings of the Demonstrations Session at EACL 2012*, Avignon, France. Association for Computational Linguistics.

Jie Yang, Yue Zhang, Linwei Li, and Xingxuan Li. 2017. Yedda: A lightweight collaborative text span annotation tool. *arXiv preprint arXiv:1711.03759*.

Mirza Yusuf, Praatibh Surana, and Chethan sharma. 2022. Hindiwsd: A package for word sense disambiguation in hinglish & hindi. In *Proceedings of The WILDRE-6 Workshop within the 13th Language Resources and Evaluation Conference*, pages 18–23, Marseille, France. European Language Resources Association.