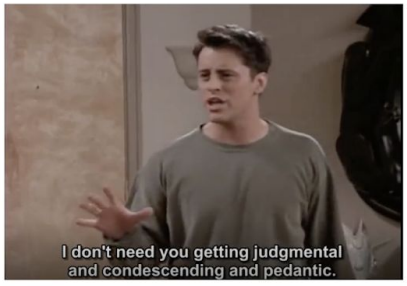
TV-Dictionary

Rahul Damineni (daminens@oregonstate.edu)

Please share a copy with everyone in the class

Your search "condescending" matched three TV videos you've watched:

Play the results to learn "condescending" in the context you know!



A regular dictionary gets you the definition of a word.

A “TV dict” gets your favourite TV characters to teach you the word:
pronunciation, usage context and a memory to remember the word – all in one go. Hit play!

User stories

User Accounts

in list User Stories

Description Edit

Users should be able to sign-up and log into their accounts.
Users should have a way to recover their account when they need it.

Acceptance Criteria:

- The login & signup forms works as intended without letting the user bypass log-in
- The session info is persisted so it provides smooth user experience while not compromising the authenticity of user actions.
- The session expires or terminates gracefully.

Content Selection

in list User Stories

Description Edit

- User should be presented a multi-select search box through which they can inform all the TV series and movies they want to see in their results.
- User should have option to access this search box to update their preferences at a later point

Acceptance Criteria:

- User could uniquely identify the intended media from the generated search suggestions as they type in
- Once the changes are made, they are persistent and can be accessed from another session

Word Lookup

in list User Stories

Description Edit

User should be presented with a search box through which they can learn a word

Acceptance Criteria:

- The search should understand misspelled words correctly
- The search should be fast enough to not bore the user
- The results should each have at least one utterance of the search term

Results Canvas

in list User Stories

Description Edit

Presents results and synonyms of the search term

Acceptance Criteria:

- The presentation should include a formal definition of the word
- The presentation should provide a way to access all matched results (example: through pagination)
- The presentation should provide hyperlinks to common synonyms of the search term
- The result canvas should have a "search again" button that resets "word lookup" feature

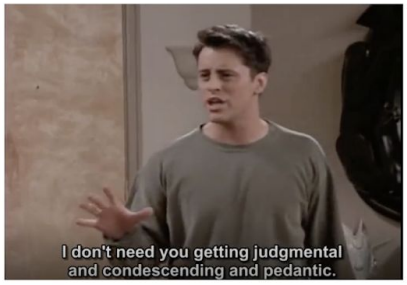
Description

A regular dictionary gets you the definition of a word.

A “TV dict” gets your favourite TV characters to teach you the word:
pronunciation, usage context and a memory to remember the word – all in one go.
TV Dict takes advantage of contextual learning and visual learning. We form abstract memories by relating them to something real. So, when you come across a new word, the idea represented by the word is better expressed by actions (taken by humans) rather some other words. Meaning, a scene helps you understand it better than a definition.

Your search "condescending" matched three TV videos you've watched:

Play the results to learn "condescending" in the context you know!



Here the word “condescending” was used in “The Big Bang Theory”, “Friends” & “True Romance”. If you had watched these before and have context around the characters, it’ll be easier to understand the true idea “condescending” represents. That’s the whole point of this system.

The core system is `word2sceneMatcher` (matches a word with specific scene in media where that word is uttered).
For that:

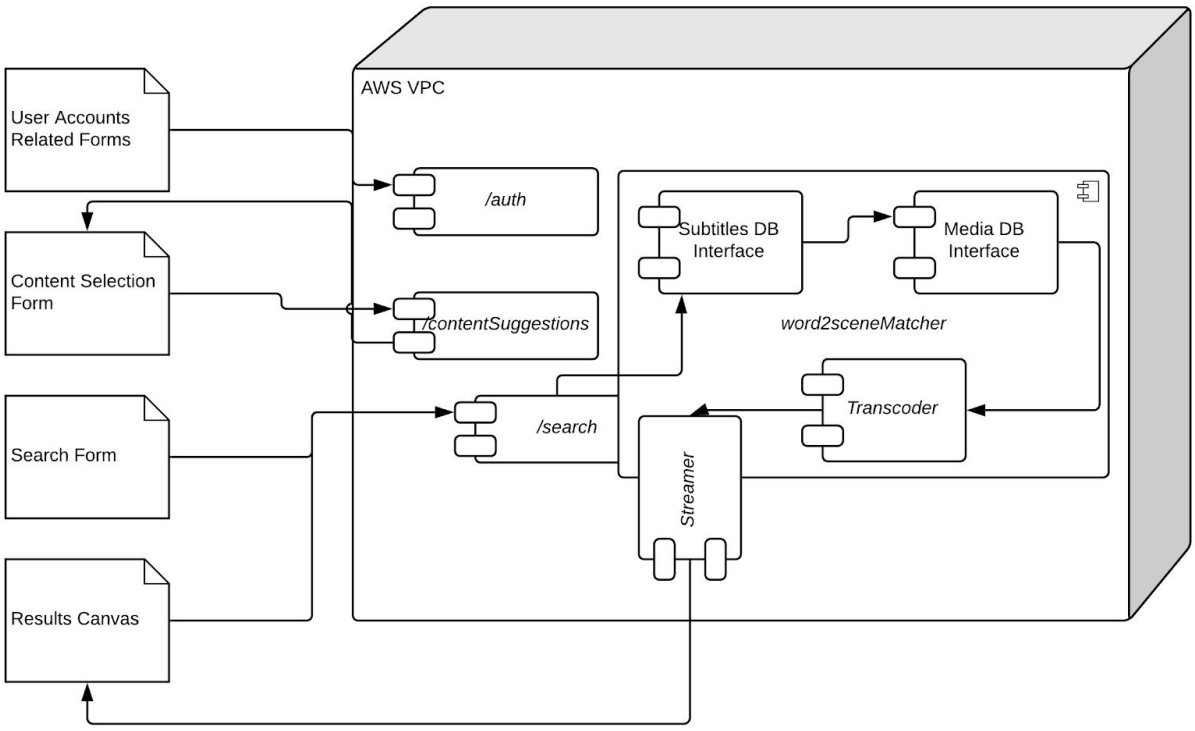
1. Collecting subtitles (closed captions) is the first step. We will use a crawler to scrape `.srt` files from popular subtitle sites like tvsubtitles.net and subscene.com. These files would have text we can search on & its time of utterance.
    - Suggested tech: `Python, Scrapy/ Selenium, Linux, AWS-EC2`
2. We’ll index the content of subtitle files and perform a full-text search to retrieve most relevant sentences (and times) from all media in our DB. This will make the search fast!
    - Suggested tech: `Elasticsearch/ PostgreSQL/ MongoDB, AWS`
3. We should pre-process the media in order to randomly access any frame at a given time and then index the raw content against its “unique identifier plus time”. We can use the results from above DB to retrieve specific scenes and stream them
    - Suggested tech: `Amazon elastic transcoder, Amazon S3 / MongoDB`

APIs & Endpoints:

1. `/auth` API for user account management. Manages sign-up, login and password recovery
2. `/contentSuggestions` endpoint for “Content Selection” story. Dynamically suggests possibilities as user types.
3. `/search` endpoint for “Word Lookup” story. Triggers the above “core system”
4. `/synonyms` endpoint for “Results Canvas”. Given a word, returns it’s synonyms.
    - Suggested tech: `ExpressJS, Node or Python, Flask`

Frontend:

1. Should be responsive.
2. Preferably a Single Page Application (SPA) without server-side rendering
    - Suggested tech: `JavaScript, React/ Angular or Python, Django/ Flask`



Component Diagram