Inter(pre|in)vention

Team

- Arunachalam Sakthivel Manikam
- Ashley S Tisdale
- Ashwin Viswanathan
- Sanil Sinai Borkar
- Thomas R Storey

Concept

Writing and reading are collaborative and social behaviors. Language is a commons that is constituted by the community of language practitioners. We literally create language when we practice it together by reading, speaking, listening and writing. Technologies of language practice, be they oral, literary, or electrate in nature, form the vectors along which language enters the public sphere, giving it the potential to change and morph. As a product of the public, language is defined by accident, happenstance, and the various confluences of existing vectors, forces and states in human systems.

How can new systems of technological and cultural production change the ways we use language? When writing and reading are fully exposed and made available by the technical apparatus, what new patterns of meaning-making can emerge? As network and computational technologies become more powerful and available, how can their prosthetic, cybernetic functions change the way we write, read, and make inscription upon the commons of language?

In this project, tentatively titled **Inter(pre|in)vention**, we propose a performative, networked model for collaborative poetic production across both human and non-human invention. We plan to take advantage of the statistical errors of computational interpretation of human reading to generate new poetry, and as a result, new ways of meaning-making. These computational errors become metaphors for intuitive leaps. The computer's misinterpretation (and the resulting poems) will be displayed as part of an ever-changing network model. Using this network visualization, we can examine the collaborative process of language potential as it is actually happening. The network mediated platform becomes a visual and tangible, operational manifestation of the always collaborative, collective process of language making.

Description

Inter(pre|in)vention is a website that prompts the user to read a poem aloud. The website harnesses the user's microphone to listen to and interpret her speech into text. Much of the time the conversion will be successful, but often it will fail, and these failures become sites of invention. Each of these failures, or rather, re-interpretations, will be stored in textual form to be used as prompts for future readers. These users can then read the re-interpretations aloud to repeat the process so that the poems are in constant collaborative revision and production. As each poem begets new poems, a network graph grows from a single origin root, and this poetic network graph will be spatially explorable as a hypertext poem.

Each newly generated poem will be compared against its parent and sibling poems to find where it differs. These differences will drive the re-interpretation of the child poems by searching for the words on the Internet and/or our database of stored poems to find related words and texts. This feedback loop, contextually reseeding each new poem, will progress/regress each branch in radically different directions.
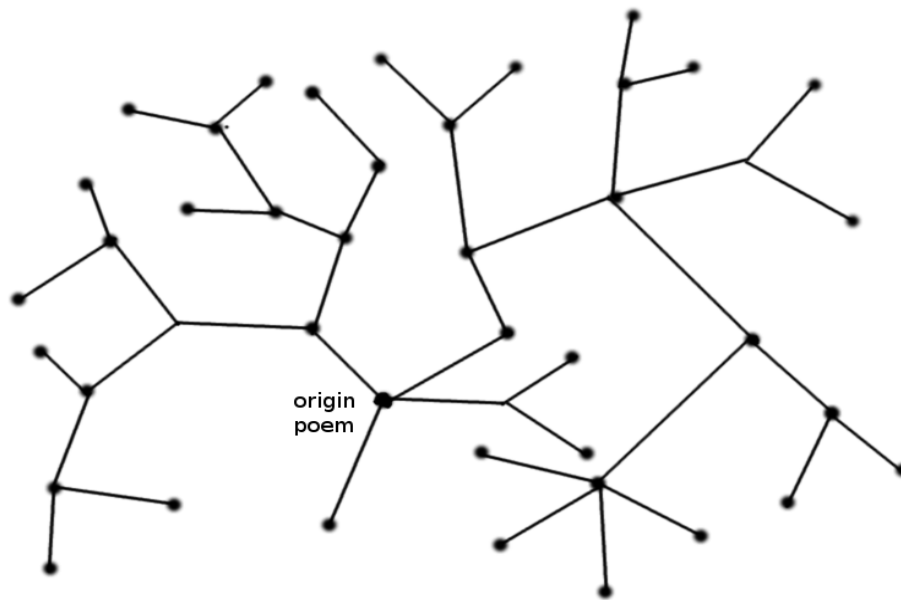
figure 1: Visual representation of the poem network graph after multiple readings of the origin poem as well as the generated poems.

As the networked, rhizomatic hypertext poem grows, new ways of reading and writing become possible. Statistical models can be observed from patterns of deviance along individual paths from origin to leaf node. For example, sibling readings can be considered together to find their common and divergent deviances from their shared parent reading. Readings at the same distance from one origin can be overlaid or juxtaposed with readings at another level. Once the poem becomes a network, it becomes non-hierarchical, and the primacy of the origin is lost in the constellation of collective making.

Our origin corpus of poetic text will be made up of selections drawn from Emily Dickinson's body of work. This decision is very specific: Dickinson was a reclusive writer who wrote largely for herself in private. It was not until after her death that her poetry was discovered and later published. By using her work as a starting point for our collaborative, collective process of language-making, we highlight the always public and contingent nature of language that even Emily Dickinson was, and is, a part of. Poetry becomes a dynamic force in language production, so that those who are private readers can also become public writers and active participants in the process of meaning-making.

Inter(pre|in)vention

Listen

An ignorance a Sunset

Confer upon the Eye
Of Territory - Color -
Circumference - Decay -

Its Amber Revelation
Exhilirate - Debase -
Omnipotence' inspection
Of Our inferior face -

And when the solemn features
Confirm - in Victory -
We start - as if detected
In Immortality -

an ignorance a sunset confer upon the high of territory color circumference decay if amber revelation exhilarate the base omnipotence inspection of their inferior face when the following features confirm and victory we start as if the kid any more challenging

figure 2: Demonstration of Web Speech API processing a reading from an Emily Dickinson text. The original text is on the left, the text produced from the API's listening to the reader is on the left. Future versions will feature more poetic formatting generated from the cadence and rhythm of the reading.

Implementation

Once the user selects one of the many poems that we offer or the already generated "diverse" ones and reads it, we display the interpretation of her poem on the screen beside the selected version of the poem. From the visualization perspective, we then highlight the words that differ from the original source and make the other words fade away. These highlighted words, as mentioned earlier, will be used to create a new poem, and will be stored as a child of the selected poem in the network graph.

For each pair of differing words, we compute their [Levenshtein's](#) [Distance](#) and use this as a directing factor in the search for new content/context to guide our data mining algorithm in a directed fashion. We then arrange the mined words in a human readable fashion to generate a new poem, which will then be stored in our database to be used by another user as her "selected" poem.

Front-end Development

The front-end (browser) will be developed in HTML5/CSS, JavaScript and Angular JS.

For visualization purposes, we will make use of JavaScript frameworks like Angular JS. For carrying out operations pertaining to the Natural Language Processing (NLP) part of our implementation, viz. computing the Levenshtein's distance, we will be using the already available JavaScript libraries instead of re-inventing the wheel.

Back-end Development

The back-end (database) will be developed in Node.js. All the data will be stored in flat-files.

Working Diagram

Poems list

Poem1

Poem2

Poem3

Poem4

Poem5

User

User selects a Poem

Graph of previously generated versions are displayed

Display selected version

Poem1

Speech Recognition API is used

Different version of Poem1

(Output of Speech Recognition API)

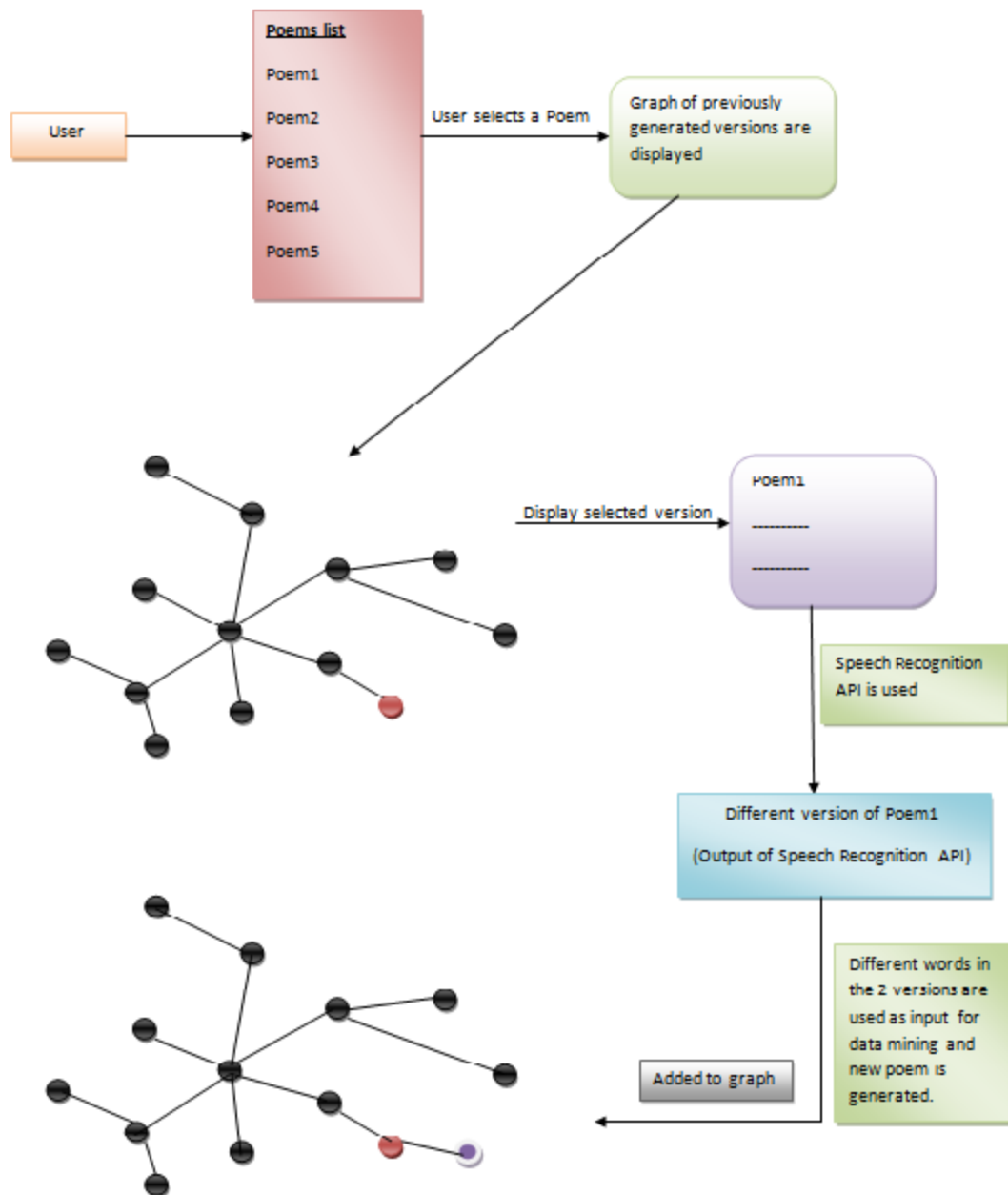Different words in the 2 versions are used as input for data mining and new poem is generated.

Added to graph

figure 3: Workflow diagram