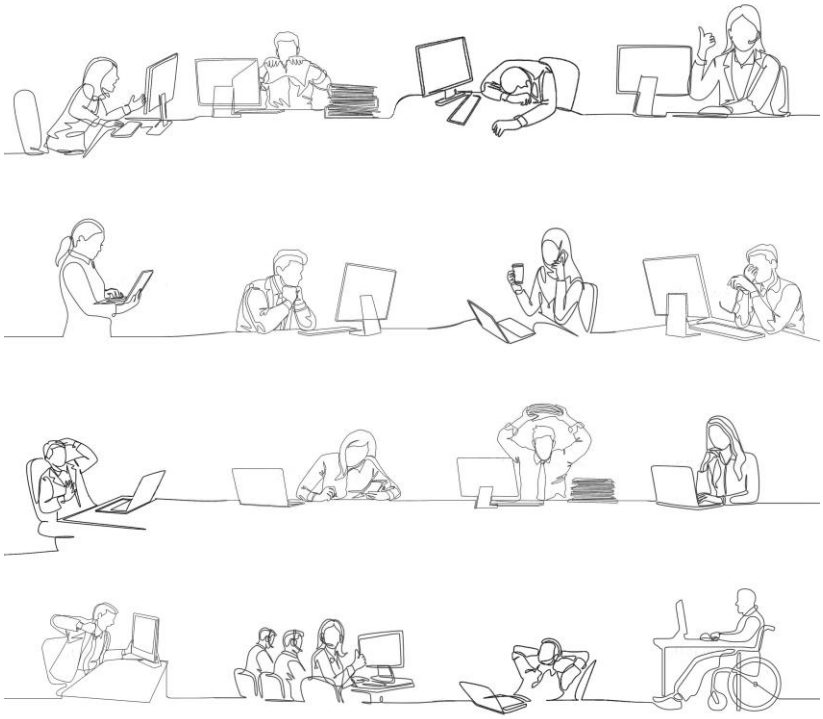
Big Sistah

Quantifying the wellbeing of multilingual remote workers in real-time



Codice	**2022EYX28N**
Duration	24 months
Main ERC field	**SH - Social Sciences and Humanities**
ERC subfields	SH3_13 *Digital social research* SH3_12 *Communication and information, networks, media* SH3_08 *Social policies, welfare, work and employment*
Keywords	information society; remote working; multilingualism; wellbeing; work and organization psychology; data analysis techniques;

Brief description of the proposal

Big Sistah aims to create and combine indicators for real-time monitoring *remote workers'* (RWs) activities at their PCs to study their (1) profiles, (2) emerging working habits, (3) mental fatigue, stress, and motivation (through attentional changes), and (4) their impact on their efficiency (e.g., multitasking), efficacy (e.g., expertise levels), and productivity.

RWs' wellbeing has been mainly studied with competing survey methods and introspective indicators, yielding scattered, hardly comparable results. Work psychology, HCI, usability, writing process studies have developed some new indicators to study RWs, but current research efforts rarely triangulate data to reliably derive new knowledge—especially from the scope of the RWs' wellbeing.

Thus, the ongoing labor revolution towards remote working is mostly based on trial and error, often with mixed results. New consulting start-ups provide companies with guidance to switch to remote working, but they tend to focus on the control of the employees and their productivity, and they generally disregard the deep changes in the RW's ways and in work setups.

The landslide shift to remote working setups, due to the pandemic, crucially opens many unique opportunities. From a labor welfare perspective, it lets us study emerging behaviors in RWs common to many job profiles, from healthcare providers to white collar civil servants. RWs add human value (e.g., expert analysis and decision making) to intensive information-processing tasks through HCI, often in multilingual settings. In 2022, more than 50% of the world's population is using the Internet ca. 7 hours a day—so, not only for leisure—and they reach contents in other languages daily. Gist machine translation is now part of people's everyday lives.

Current tools to measure new indicators are often proprietary software, mainly for RWs' surveillance. Open-source prototypes are too generic, and combining them leads to clunky, unrealistic, and unreliable settings and results. Furthermore, social research inspired by situated cognition, like Big Sistah, demands non-invasive methods to access RWs' performance in their natural work environments and with full respect to their privacy.

A key contribution of our project is to develop the technology to seamlessly collect data at runtime, to empower the scientific community with an open-source, interdisciplinary research toolbox to collect and measure RW's data in real time. The software will be made available as an open platform, for many disciplines to use and enrich with further indicators, now with a common base. Big Sistah aims to become the standard to support labor guidelines and best practice recommendations to ensure that economical viability is not reached against citizens' welfare.

Basic BigSistah research workplan

The tool/library suggestions below are for orientation and are not meant to be restrictive.

Tasks

1. Preliminary analysis and set-up. 2 weeks

1.1. Project requirement gathering.
1.2. Selection of development tools and frameworks.
1.3. Initial setup of development environment.

Suggested tools: Git for version control, JIRA for project management, Visual Studio Code or JetBrains IDEs for development.

2. Keylogger, text diff, and internet tracking development. 6-8 weeks

2.1. Keylogging functions capturing keydown, keyup, and keypress events in Unicode. 1-2 weeks
Libraries: InputSimulator for C#, pynput for Python
2.2. Mouse event tracking with millisecond-level timestamping. **1** week
Libraries: MouseEvent for C#, pyautogui for Python
2.3. Text diff functionality for text changes for !ME-supported languages and backtracking. 1 week
Libraries: diff-match-patch for C#/Python
2.4. Implement Internet tracking for recording online searches. 1-2 weeks
Libraries: Selenium WebDriver for browser automation, WebRequest in C#, Requests in Python

3. Database architecture and integration 2-3 weeks

3.1. Design database schema for SQL database. 1 week
Tools: MySQL Workbench, PostgreSQL with pgAdmin, SQLite
3.2. Integration of database with keylogging and other data collection modules. 1-2 weeks
Libraries: Entity Framework for C#, SQLAlchemy for Python

4. Survey and psychological testing module 5-7 weeks

4.1. Implement informed consent collection. **1** week
4.2. Develop modules for collecting ad hoc surveys. 1-2 weeks
Libraries: SurveyJS for online surveys, Qt for offline surveys
4.3. Implement psychological and language-related tests (approx. 12 tests). 2-3 weeks
Libraries: psychopy for Python-based psychological tests

5. Data analysis app 4-9 weeks

5.1. Text preprocessing for analysis 1-2 weeks
5.1.1. Text cleaning
Libraries: BeautifulSoup for HTML/XML stripping; Python's re module for other regular expressions.
5.1.2. Tokenization
Libraries: NLTK, spaCy
5.1.3. Part-of-speech tagging
Libraries: NLTK, spaCy, Stanford NLP

5.1.4. Frequency count
Libraries: Python's built-in collections.Counter or pandas for more complex data manipulations.
5.1.5. Named entity recognition
Libraries: spaCy, Stanford NLP
5.1.6. Sentiment analysis
Libraries: TextBlob, NLTK (SentiWordNet), or pre-trained models in TensorFlow or PyTorch
5.1.7. N-gram formation
Libraries: NLTK, TextBiob, or custom Python code.
5.1.8. Discourse marker identification
Libraries: Custom code, possibly utilizing tokenization and POS tagging.

5.2. Integration with SQL database. 1 week
Libraries: Entity Framework for C#, SQLAlchemy for Python

5.3. Text alignment aligning source/source, source/target, and target/target texts). 1-2 weeks
Libraries: NLTK or spaCy for text alignment

5.4. Development of statistical analysis and visualization tools 1-2 weeks
Libraries: pandas, Matplotlib, Seabornfor Python-based data analysis

Additional tasks

Testing and debugging: 12 weeks
Documentation: 4 weeks
Optimization: 16 weeks

Total time, 18 months
(Buffer time, 6 weeks)

Nota bene

1. Work it to be done in presence at the Forli office (Italy).
2. English B2 is a must, Italian not required but very welcome. Other languages welcome as well.
3. The application is to be developed as a team. Another early-stage researcher will be active at Milano-Bicocca, working on screen reording, webcam recording, audio system and mike recording, and the GUI. Two profs of computing are at Milano-Bicocca and two psychology professors are at the campus ofUnibo in Cesena. You are expected to coordinate with everyone in the team, but also to be able to work independently.
4. Rough time estimates do not include supervision, team meetings and some buffer time. The rest of the time will be devoted to implement improvements, debug, tweak details and enalrge and improve analyses. The project is initially planned to be developed in one year (12 months) plus *at least* 4.5 months for testing and publications.